Text Generation

Karine Baschung, Gabriel G. Bès, Denis Carcagno, Corinne Fournier

Text Generation

Karine Baschung [1], Gabriel G. Bès [1], Denis Carcagno [2], Corinne Fournier [3]

1. GRIL, Université Blaise-Pascal, Clermont-Ferrand
2. Alcatel Alsthom Recherche
3. Dassault Aviation

Report for HYPERDOCSY, ESPRIT Exploratory Action 5652, Task 2, October 1991.

Abstract

The Hyperdocsy ESPRIT Exploratory Action (11/1990-09/1991) studied the automatic production of technical documentation. The documentation of the project includes three main parts: Domain Modelling Analysis (task 1), Text Generation (task 2) and Outline of a Future Hyperdocsy System (task 3). This document is the report on task 2, Text Generation. It is organized in two main parts. The first part deals with corpus analysis, with a view to identify the obstacles to be dealt with when generating text. Different aspect of the corpus are analyzed: topics and rhetorical structure, communicative organization, syntax, semantics, lexicon. The second part deals with the evaluation of existing text generation techniques. It compares two models: Unification Categorial Grammar (UCG) and the Meaning-Text Theory (MTT), and proposes general criteria for evaluating text generation systems.

HYPERDOCSY

ESPRIT EXPLORATORY ACTION NO 5652

REPORT

TASK NO 2

Text Generation

Prepared by:

Karine Baschung, Universite Blaise Pascal, Clermond Ferrant
Gabriel Bes, Universite Blaise Pascal, Clermond Ferrant
Denis Carcagno, Alcatel Alsthom Recherche
Corinne Fournier, Dassault Aviation

October 1991

Text Generation

Table of Content

1. Introduction

The two objectives of Task 2 on Text Generation are the following:

- The analysis of documents on the domain chosen in Task 1 and the identification of obstacles to be dealt with when generating text;

- The evaluation of existing text generation techniques.

This report on text generation contains two chapters corresponding to the main subtasks of Task 2: corpus analysis and evaluation of existing generation techniques.

Corpus analysis helps to determine the specificities and problems encountered in the document on the PMS. The results of the corpus analysis are then an input to the evaluation of existing generation techniques and a first specification of linguistic knowledge needed in order to generate the documentation.

Different aspects of the corpus are analyzed:

- Topics and rhetorical structure
- Communicative organization
- Syntax
- Semantics
- Lexicon

The topical analysis of the text reveals its primitive units of information, called topics. Primitive topics combine into more complex ones and thus constitute the text structure of the whole document.

The organization of sentences and texts fulfils the communicative goal the speaker or writer has in mind. A number of linguistic devices may be used to realize different communicative effects. Five aspects are examined: communicative organization proper, distribution of information, cohesion, thematic progression and the relation between the topical structure and the thematic structure.

The main syntactic constructions found in the text are identified. Ambiguities due to some of these constructions are pointed out.

The main semantic phenomena are identified.

The format of a dictionary entry is proposed so as to describe some lexical items of the corpus.

The structure of the report is the following: methods for analysis and results are in the body of the document while data from corpus analysis are in appendices.

Evaluation of existing text generation techniques consists in the comparison of two models: UCG (Unification Categorial Grammar) and MTT (Meaning-Text Theory).

This chapter includes:

- A presentation of both models, including the representation of a sample sentence;

- An evaluation of both models according to the results of the corpus analysis so as to see whether the problems encountered in the text may be solved or not;

- An identification of general criteria for evaluating in general text generation systems independently of application type and domain. These criteria are used to evaluate both models;

- The choice of the model meeting all requirements (corpus analysis and general criteria) for the future HYPERDOCSY system.

Section 2.1 of this report was written by Alcatel Alsthom Recherche, section 2.2 by Dassault Aviation, sections 2.3 -2.7 by Clermont-Ferrand. Sections 3.1.1, 3.2.1 and 3.3.1 were written by Clermont-Ferrand, sections 3.1.2, 3.2.2 and 3.3 by Alcatel Alsthom Recherche and Dassault Aviation.

The task leader responsible for the planning and layout of the report was Dassault Aviation.

2 CORPUS ANALYSIS

2.1 Topics and rhetorical structure

2.1.1 Preamble

The goal of this paper is to set a common approach in analyzing the corpus from rhetorical and topical points of view. That common approach is needed as we want to be able to coordinate the different partner works performed on that task. Furthermore, results obtained, adopting this approach, will lend themselves to a smooth integration into a common knowledge base valuable for text planning and conceptual model in our future project.

2.1.2 Rhetorical structure and topical coverage

We assume that documentation text is organized rationally in different parts. Each of these parts whose purpose is to achieve a given communicative goal is organized recursively in sub-parts, owning their proper goals, which participate in the achievement of the overall goal. At the last level of this recursive structure elementary units of communicative information can be found which can be viewed as primitive questions, i.e. questions that cannot be decomposed into simpler ones. These elementary units are called topics ([Carcagno, 89]). Topics present an additional property. Each of them represents a query for information addressed to the conceptual model of the domain and thus defines the detail to be achieved in the model in order to satisfy the generation of information. Therefore each topic must be equipped with a function which completed with right arguments returns a message which will constitute an element of the final text.

This approach advocates a top down analysis of the document. In the example (cf next section), top level parts defined by titles of sections are decomposed into sub-parts recursively until topical units are reached. Several difficulties are encountered during that decomposition process.

Some sequences of parts do not seem coherent or do not follow other sequences of similar parts. That may be because the content found in the domain model is so different for different parts of the text that it imposes a modification of the structure. It may also be that the text structure is not coherent. In that last case, we will propose to our domain experts an alternative structure to be validated. In the former case, must be kept in mind that the text structures we try to extract are initial patterns which organize the content extracted from the conceptual model. Thus, content extracted or subsequent text planning processes may modify initial text structure quite largely. It

will be a future work to define how these modifications operate. Only initial text structures that drive the content extraction from the domain model are of interest now.
Finally, it should be noted that purposes of big sections are defined in the standard of document used to write the document.
Part1 is an overview of structure and functions of the PMS. Part 2 focuses more in detail on the functional description of the PMS. Part 3 describes in detail the PMS graphical interface (Meanings of icons and actions associated to them). That section poses peculiar problems to text generation since it is mainly composed of panels containing fragments of sentences. At a first glance, it seems that information expressed is coming directly from design databases. If this is right, we should check whether it is possible to format that data directly in Database report forms. Part4 describes in a very systematic way the software architecture and detailed functionalities of each modules. Part 5 summarizes the interfaces between different components of the PMS. Part 6 describes test procedures to be applied to the PMS from the operator's console.

In order to get a more concrete idea on how we did proceed in making our topical analysis, let us comment an example.

The Power Management System (PMS) monitors and controls electricity production via four Generating Sets (GS), that is three Diesel Generators (DG) and one Shaft Generator (SG).

A complex sentence such as the one presented contains several messages. Each message brings a new information which can be viewed as the answer to an elementary and implicit question. In our framework, a topic corresponds to a question, a comment to the new information and the pair topic/comment to the resulting message.

According to this, the sentence above has been decomposed in two messages :

(1) *The Power Management System (PMS) monitors and controls electricity production .*
(2) *The Power Management System is composed of four Generating Sets (GS), that is three Diesel Generators (DG) and one Shaft Generator (SG).*

According to our framework, those messages are answers to questions which define our topics. Now the problem amounts to figuring out the appropriate question which could determine a given answer. Someone could argue that this problem is non deterministic, i.e. several different questions can provide the basis for a given answer. According to this point of view, the message (1) could originate from several questions :

What does the PMS do ?
Who or what monitors and controls what?
What is done with electricity production ?
Who or what monitors and controls electricity production ?
What does the PMS do to electricity production ?
...

We do think that these questions are not equally probable in the context of our message. Therefore, it should be possible to set up an order of relevance on this list of questions according to communicative constraints[1] on question/answer sequences. The first constraint relies on the fact that a question contains a focus which will define the theme of the answer (leftmost elements of the answer) :

*What **does the PMS do** ? The PMS monitors and controls...*

The second constraint is that this focus must be known to the reader or has to be introduced before in the text.

According to these constraints we are able in most cases to isolate the question that fits the message in the most relevant way. In our example, that question is the first of the list.

*What **does the PMS do** ?* gives the topic **Functions(PMS)**

Functions of the PMS are described in the text and in the domain model at different levels of detail. The functions reported here are globally described and therefore our topic will ask for the **Level1-functions(PMS)** in the model.

This method applied to our two messages will result in the following decomposition:

(1) Topic : **Level1-functions(PMS)** / Comment : **Monitoring and control of electricity production.**

(2) Topic : **Level1-structure(PMS)** / Comment : **4 GS, i.e. 3 DG and 1 SG.**

Our example reveals a problem which we have not addressed yet. Rules that allow for combining several messages (1 and 2) into one sentence are in some cases still unclear. This research question is outside the scope of this limited study and will be investigated extensively later in the project. An approach to be considered is described in ([Man 87]). For the moment, the analysis will be focused on the basic structures of the text and the knowledge in the domain model which is required by text generation.

2.1.3 Document analysis

This section contains a result of the analysis obtained by applying our method. We chose as example of text the two first parts of the design specification document. The following sections reflect the different steps of the analysis. First, the text is splitted into messages to which are assigned topics organized in rhetorical structures. Second, rhetorical structures and topics present in the text are extracted in order to get a summarized view of the text structures. This second step will help us in capturing regularities present in the text structures. Third, this summarized description is generalized. The third step provides a first output of the analysis in a form of **discourse grammar** which rules out in a productive way the underlying structures of this kind of document. Finally, meanings of topics are defined. Those definitions specify the minimal knowledge which must be expressed in the domain model in order to generate the document.

---

[1] A detailed description of the notions used to define communicative constraints for generation is reported in the *Study of communicative progression* performed during the project.

2.1.3.1 Example

This example aims at illustrating the method of analysis. Text structures will probably need to be revised, refined, also tuned according to the conceptual model. Predicates written in smaller font designate topics whereas those in bigger font designate rhetorical structures. The full analysis for part 1 and part 2 of the document is provided in the annex.

1 Introduction.

Overview(PMS)
1.1 Purpose and Scope of Power Management System.
Level1-functions(PMS)
The Power Management System (PMS) monitors and controls electricity production
Level1-structure(PMS)
via four Generating Sets (GS), that is three Diesel Generators (DG) and one Shaft Generator (SG).

Overview-main-components(PMS)
Level1-structure(SG)
The SG is connected to the Main Engine (ME)
Level1-functions(SG)
and it can produce power to either the busbar or the Bow-/Stern-Thruster (BT,ST).
Level1-structure(DG)
()
Level1-functions(DG)
()
Level2-functions(DG)
The DG part of the system is a standardized full-automatic start/stop, synchronization, frequency control, loadsharing and black out start system.
Level2-functions(SG)
The SG part include synchronization to busbar (BB) and automatic connection of SG to BT/ST.

Structural-description(PMS)
1.2 Overview of the controlled components/system.

Ref-to(figure1.a)
In figure 1.a is shown schematic the controlled/monitored system.
List-of-components(PMS)
This include the GS, the GS Main Breakers (MB), the BT/ST MB's and the Emergency Switch Board (ESB).
Level1-functions(PMS)---------> should be in 1.1
Furthermore the PMS monitors alarms from the alarm system, all alarms detected by the PMS system and information for the DG surrounding machinery.

Functional-description(PMS)
2 Functional description for normal use.

Overview(control)
2.1 Control modes in general.

Number(control-modes)
The PMS contains three modes of operation for DGs and three modes of operation for the SG - they are explained briefly below:
MANUAL:
Command-device(control-modes)
Each GS has a MANUAL/AUTO selector.
Effect(device)
When MANUAL mode is selected it overrides the two other modes AUTOMATIC and SEMIAUTOMATIC.

Description(control-modes)
Description(manual-mode)

Actions(manual-mode,DG,PMS)
DGs: *No control at all of DG in question.*
Actions(manual-mode,SG,PMS)
SG: *MB to BT,ST:* *No control at all of thruster MB in question.*
MB to Busbar (BB): *No control at all of MB in question.*

Description(auto-mode,DG)
The next two modes only concern operation of DGs. These modes are common modes for all DGs.
The modes require, that the DGs are in AUTO mode (not MANUAL).

Description(SA-mode,DG)
DG SEMIAUTOMATIC:
actions(SA-mode,DG,PMS)
The PMS will automatically perform the following functions:

1 *Black out start.*
2 *Loadsharing and frequency control of online DGs.*
3 *Only one start attempt in case of starting failure.*
4 *Synchronization, when the diesel engine is started.*

actions(SA-mode,DG,operator)
Start and stop of DGs, except during black out start, is commanded by the operator.

Description(A-mode,DG)
DG AUTOMATIC:
actions(A-mode,DG,PMS)
The PMS will automatically perform the functions 1 - 4 described for DG SEMIAUTOMATIC-mode, and the following functions:
1 *Start and stop of DGs based on actual power requirements.*
2 *Change to the next DG in the standby sequence, if a DG does not start.*

3 Start of standby DG and shut down of faulty DG on AE prewarnings.

4 Start of one or two DGs (load dependent), when SG is wanted stopped either because mode is changed to a mode without SG on the ship handling mode selector (ref.[2]) (SG AUTOMATIC mode only) or by command from the ISC consoles (SG SEMIAUTOMATIC mode only).

5 Start of two DGs if SG online has a standby start shut down upon ME slowdown or if SG frequency is above/below allowed range for BB operation.

actions(A-mode,DG,operator)

()

Description(auto-mode,SG)

The next two modes only concern operation of SG.

Relationship-between(DG,SG,Auto-mode)

Operation of DG is independent of selected mode SG SEMIAUTOMATIC and SG AUTOMATIC.

Description(SA-mode,SG)

SG SEMIAUTOMATIC:

actions(SA-mode,SG,PMS)

The PMS will perform the following functions:

1 Synchronization of SG to BB.

2 Immediately stop of DGs online after SG MB to BB is closed.

3 Start sequence for switching BT/ST online.

4 Stop sequences for switching SG off-line from either BB or thruster.

actions(SA-mode,SG,operator)

Start and stop of SG's to either BB or BT/ST is commanded by the operator.

Description(A-mode,SG)

SG AUTOMATIC:

actions(A-mode,SG,PMS)

The PMS will automatically perform the functions 1 - 4 described for SG SEMIAUTOMATIC-mode, and the following functions:

1 Automatic control of SG to either BB or BT/ST dependent of mode selected on ship handling mode selector (ref[2]).

actions(A-mode,SG,operator)

()

Operations(control)

2.2 PMS operation strategy.

Operations(control,DG)

Action-description(Blackout-start,DG)

Conditions(Blackout-start,_,_,DG,_)

Blackout start is enabled when at least one DG is in AUTO-mode and not blocked

Definition(blocked,DG)

(blocked means that the DG is not available f.x. because of an alarm).

level1-process(Blackout-start,DG)

One of two actions will take place after a blackout:

1 If one or more DGs are running the highest priority will be switched online when its frequency has reached a preset level.

2 If no DG is running, the first in the standby sequence will be started and switched online, when its frequency has reached a preset level.

The next DG in the standby sequence will be started if the former DG fails to start or switch online.

Definition(switch-online,DG)

Switch online means in this case direct connection without synchronization of MB to BB commanded by the PMS system...

2.1.3.2 Text Structures extracted from PMS document

This section contains a complete description of the actual text structure. From this material, our final aim is to extract recurrent structures. These structures, which are called rhetorical structures or rhetorical schemas ([McKeown, 85]), are used to drive the determination of the document content from the domain model.

Overview(PMS)
- Level1-functions(PMS)
- Level1-structure(PMS)
- Overview-main-components(PMS)
  - Level1-structure(SG)
  - Level1-functions(SG)
  - Level1-structure(DG)
  - Level1-functions(DG)
  - Level2-functions(DG)
  - Level2-functions(SG)

Structural-description(PMS)
- Ref-to(figure1.a)
- List-of-components(PMS)
- List-of-connections(PMS)

Functional-description(PMS)

- Overview(control)
  - Number(control-modes)
  - List(control-modes)
  - Command-device(control-modes)
    - Effect(device)

Description(control-modes)
    Description(manual-mode)
        Actions(manual-mode,DG,PMS)
        Actions(manual-mode,SG,PMS)
    Description(auto-mode,DG)
        Description(SA-mode,DG)
            Actions(SA-mode,DG,PMS)
            Actions(SA-mode,DG,operator)
        Description(A-mode,DG)
            Actions(A-mode,DG,PMS)
            Actions(A-mode,DG,operator)
    Description(auto-mode,SG)
        Relationship-between(DG,SG,auto-mode)
        Description(SA-mode,SG)
            Actions(SA-mode,SG,PMS)
            Actions(SA-mode,SG,operator)
        Description(A-mode,SG)
            Actions(A-mode,SG,PMS)
            Actions(A-mode,SG,operator)

Operations(control)
    Operations(control,DG)
        Action-description(Blackout-start,DG)
            Conditions(Blackout-start,_,_,DG,_)
            Definition(blocked,DG)
            Level1-process(Blackout-start,DG)
            Definition(switch-online,DG)
        Action-description(priority-decision,DG)
            Conditions(priority-decision,_,_,DG,_)
            Level1-process(priority-decision,DG)
            Summary(priority-decision,DG)
    Operations(control,SG)

Level1-description(control)

    level1-description(control,DG)
        agents(control,DG)
        command-device(control)
            effect(device)
        level1-description(control,manual-mode,DG,_)
            agents(control,manual-mode DG,_)
            level1-actions(control,manual-mode,DG,_)
        level1-description(control,auto-mode,DG,_)
            agents(control,auto-mode DG,_)
            level1-actions(control,auto-mode,DG,_)
            level2-description(control,SA-mode,DG,_)
                agents(control,SA-mode,DG,_)
                functions(control,SA-mode,DG,operator,_)
                functions(control,SA-mode,DG,PMS,_)

Alarm-rules(SA-mode,DG)
Level1-actions(PMS, DG,SA-mode,_)
Level1-actions(operator,DG,SA-mode,_)
Definition(stopping)
Definition(starting)
level2-description(control,A-mode,DG,_)
agents(control,A-mode,DG,_)
functions(control,A-mode,DG,PMS,_)
Level1-actions(PMS, DG,A-mode,_)
Operation-rules(control, DG, auto-mode)
Alarm-rules(auto-mode,DG)

level1-description(control,SG)
Overview(control,SG)
Agents(control,SG)
command-device(control)
effect(device)
level1-description(control,manual-mode,SG,BB)
agents(control,manual-mode SG,BB)
level1-actions(control,manual-mode,SG,BB)
Process(synchronization,manual-mode,operator,DG,BB)
level1-description(control,auto-mode,SG,BB)
agents(control,auto-mode SG,BB)
level1-actions(control,auto-mode,SG,BB)
Process(synchronization,auto-mode,PMS,SG,BB)
level1-description(control,manual-mode, SG, Thrusters)
agents(control,manual-mode SG,Thrusters)
level1-actions(control,manual-mode,SG,Thrusters)
Process(voltage-control,manual-mode,operator,SG,Thrusters)
level1-description(control,auto-mode,SG,Thrusters)
agents(control,auto-mode SG,Thrusters)
level1-actions(control,auto-mode,SG,Thrusters)
Process(_,_,auto-mode,Thrusters)
Alarm-rules(auto-mode,Thrusters)
level2-description(control,SA-mode,SG,BB)
agents(control,SA-mode,SG,BB)
functions(control,SA-mode,SG,operator,BB)
process(connection,SA-mode,operator,SG,BB)
conditions(connection,SA-mode,operator,SG,BB)
process(disconnection,SA-mode,operator,SG,BB)
conditions(disconnection,SA-mode, operator, SG, BB)
functions(control,SA-mode,SG,PMS,BB)
level2-description(control,SA-mode,SG, Thrusters)
agents(control,SA-mode,SG,Thrusters)
functions(control,SA-mode,SG,operator, Thrusters)
process(connection,SA-mode,operator,SG, Thrusters)
conditions(connection,SA-mode,operator,SG, Thrusters)
process(disconnection,SA-mode,operator,SG, Thrusters)
conditions(disconnection,SA-mode,operator,SG, Thrusters)
functions(control,SA-mode,SG,PMS,Thrusters)

level2-description(control,A-mode,SG,_)
agents(control,A-mode,SG_)
functions(control,A-mode,SG,operator,_)
functions(control,A-mode,SG,PMS,_)
process(connection,A-mode,PMS,SG)
conditions(connection,A-mode,PMS,SG)
process(disconnection,A-mode,PMS,SG)
conditions(disconnection,A-mode,PMS,SG)
Level1-actions(PMS, SG,A-mode)
Operation-rules(control, SG, auto-mode)
Alarm-rules(auto-mode,SG)

Overview(power-reservation)
Agents(power-reservation,consumer)
process(power-reservation,_,PMS,consumer,_)
level1-process(power-reservation,consumer)

2.1.3.3 Text patterns present in the PMS document (parts 1 and 2)

From the material extracted in the previous section, we propose a first version of the underlying text structures that governs content extraction from domain model. In addition, we will have to define in future work the rhetorical relations that hold between topics. That would allow us to compute dynamically the final structure of the text.

The basic text structures are expressed by patterns which serve as rules for guiding the determination of content. Thirteen rules constitute the discourse grammar reflected by our text. Each rule lists, in its body, the topics and rhetorical structures which must be instantiated in order to satisfy a given rhetorical structure assigned to its head.

1
Overview(**component**) <=
Level1-functions(**component**)
Level1-structure(**component**)
Overview-main-components(**component**)

2
Overview-main-components(**component**) <=
for each sub-$component_i$ of **component** :
Level1-structure($component_i$)
Level1-functions($component_i$)
Level2-functions($component_i$)

3
Structural-description(**component**) <=
Ref-to(figure)
List-of-components(**component**)
List-of-connections(**component**)

4
Functional-description(**component**) <=
for each **function**$_i$ of **component** :
Overview(**function**$_i$)

5
Overview(**function**) <=
Number(**function**-modes)
List(**function**-modes)
Command-device(**function**)
Description(**function**-modes)
Operations(**function**)
Level1-description(**function**)

6
Description(**function**-modes) <=
for each **mode**$_i$ of **function**:
Description(**mode**$_i$)

7
Description(**mode**) <=
for each **object**$_i$, **sub-mode**$_j$, **agent**$_k$ of **mode**:
Actions(**sub-mode**$_j$,**object**$_i$,**agent**$_k$)
Relationship-between(**object**$_i$,**object**$_{i+1}$,**sub-mode**$_j$)

8
Operations(**function**) <=
for each **object**$_i$, **action**$_j$ performed during **function** :
action-description(**action**$_j$,**object**$_i$)

9
Action-description(**action**,**object**) <=
Conditions(**action**,**mode**,**agent**,**object**,_)
Level1-process(**action**,**object**)

10
Level1-description(**function**) <=
for each **object**$_i$ on which **function** is applied:
level1-description(**function** ,**object**$_i$)

11
level1-description(**function** ,**object**) <=
Overview(**function** ,**object**)
Agents(**function** ,**object**)
Command-device(**function**)
for each **mode**$_i$ of **function** and other relevant **object**$_j$ involved :
level1-description(**function**,**mode**$_i$,**object**,**object**$_j$)

12

level1-description(**function,mode,object,object$_j$**) <=
agents(**function,mode,object,object$_j$**)
level1-actions(**function,mode,object,object$_j$**)
for each **sub-mode$_i$** of **mode :**
level2-description(**function,mode$_i$,object,object$_j$**)
for each **action** of level1 found out above :
Process(**action, mode,agent,object ,object$_j$**)
Operation-rules(**function, object, mode**)
Alarm-rules(**mode,object$_j$**)

13

level2-description(**function,mode$_i$,object,object$_j$**) <=
agents(**function,mode$_i$,object,object$_j$**)
for each **agent** found out above :
functions(**function,mode$_i$,object,agent ,object$_j$**)
Level1-actions(**agent , object,mode$_i$,object$_j$**)
Definition(concept) when concept is introduced above.
for each **action** of level1 found out above :
process(**action ,mode$_i$,agent ,object,object$_j$**)
conditions(**action,mode$_i$,agent ,object ,object$_j$**)
Alarm-rules(**object,mode$_i$**)

2.1.3.4 Definitions of topics

For the moment, we give to each topic a first definition which needs to be tested against our first domain model and refined accordingly. This provides the modelling task with a preliminary specification of requirements.

Level1-functions(component) : looks for the global functions of component.

Level1-structure(component) : looks for the main components of component.

Level2-functions(component) : looks for the functions of component.

Ref-to(figure) : gets the content of figure.

List-of-components(component) : lists the sub-components of component.

List-of-connections(component) : lists the connections between sub-components of component.

Number(F) : returns the number of concepts of type F.

List(F) : lists the concepts of type F.

Command-device(F) : looks for the device that allows for commanding F.

Actions(M,O,A) : looks for the actions performed by A on O in mode M.

Relationship-between(O1,O2,M) : returns the relationships between objects O1 and O2 in mode M.

Overview(F,O) : Which global operations are executed when applying function F to object O ?.

Agents(F,O) : lists the agents of function F applied to object O.

Agents(F,M,O1 ,O2) : Returns the agents of function F applied to O1 and O2 in mode M.

Level1-actions(F,M,O1 ,O2) : looks for detailed actions performed on O1 and O2 in mode M when applying function F.

Operation-rules(F, O, M) : looks for rules appliable during execucution of function F on object O in mode M.

Alarm-rules(M,O1) : looks for the alarm rules about object O in mode M which can be fired.

Functions(F,M,O1 ,A ,O2) : sub-functions of F performed on O1 and O2 by A in mode M.

Level1-actions(A, O1, M, O2) : detailed actions performed by A on O1 and O2 in mode M.

Definition(concept) : fetch the concept meaning.

Process(Act ,M,A ,O1 ,O2) : returns the sequence of operations to be achieved when agent A performs action Act on objects O1 and O2 in mode M.

Conditions(Act ,M,A ,O1 ,O2) : returns the conditions that ought to be verified in order for agent A to perform action Act on O1 and O2 in mode M.

Level1-process(Act ,O) : returns the detailed description of the sequence of operations needed in order to perform action Act on object O.

2.1.4 . Conclusion

The goals of the study were to :

- Develop a method able to describe the structure of large pieces of text and reveal the difficulties encountered during that process.
- Apply that method in order to effectively produce a textual description of a significant sample of text.
- Provide requirements about the knowledge needed in the domain model in order to be able to generate such documents.

To our knowledge, there are few theories which provide guide-lines to extracting text structures. Apart from rare exceptions ([Mann 87]), they are usually applied in order to analyze small sample of texts. Finally, none of them have been proved to be computationally tractable without further developments. Our contribution goes in that direction and while there is still work to be done in that area, most of the points above have been reached.

First of all, the method reveals applyable to our kind of texts. A model of grammar has been produce which gives us the ability to structure technical descriptions of an equipment. Another result consists of the list of topics which specifies part of the knowledge required in the domain model.

Among the requirements put on future work, we should notice some of them.

The Power Management System is described recursively in a more and more detailed way in the document. This suggests that we will have a multi-layer domain model, each layer describing the same reality with a different degree of detail.

Many topics result in a message which expresses a rule or a set of rules. We will have to find out the proper way to represent rules in our domain model.

Topics named process introduce parts of text that do not seem to have a general rhetorical structure. They correspond to description of sequences of events and actions. Thus their structure follows a pattern determined by the sequence itself, i.e (1) event x... (2) participants y, z in that event... (3) following actions of y...
These patterns are domain dependant and can be modelized using the notion of domain dependant topic tree (see [Carcagno, 89]).

Definitions of technical notions are spreaded over the document, taking place usually when they are used for the first time.

2.1.5 References

Bourbeau L., Carcagno D., Goldberg E., Kittredge R.,Polguère A.: *Bilingual Generation of Weather Forecasts in an Operations Environment,* Coling 90, Helsinki, Août 1990.

Carcagno D., Iordanskaja L.: *Content Determination and Text Structuring in GOSSIP,* Second European Workshop on Natural Language Generation, Edinburgh, Avril 1989.

Mann W., Thompson S., 1987.: *Rhetorical Structure Theory: A Theory of Text Organization,* Technical Report Information Sciences Institute, Marina del Rey, June 1987.

Mc Keown K., 1985.: Text Generation, Cambridge University Press, Cambridge, 1985.

2.2 THE COMMUNICATIVE ORGANIZATION

2.2.1 SURVEY OF THE NOTION

2.2.1.1 INTRODUCTION

2.2.1.1.1 THE VARIOUS APPROACHES

The notion of THEME raises a lot of problems. The terminology, the relation of this notion to other notions, and the role of the theme itself vary a lot depending on the theory. This introduction reuses the presentation of a special issue of *Langue francaise* on the theme [Cadiot, 1988].

Here are the different theories or approaches which tackled the notion :

- Theoretical syntax
  - Modelling of competence
  - Description without formalization
  - Grammars of various languages (mostly German & Slavic languages)
- Descriptive linguistics
  - Specific phenomena in exotic languages
  - Typology of languages
  - Word order
- Pragmatics
  - Presupposition
  - Enunciation
  - Argumentation
- Computational linguistics
  - Discourse analysis
  - Text generation
- Discourse analysis, Literary criticism, Exegesis

2.2.1.1.2 THE ROLES OF THE THEME AND RHEME

According to the theory, the theme is seen as:

- A syntactic position
- An actancial role
- The part which is presupposed
- The part with a weak communicative function
- A psychological focus center
- Acohesion device
- A relevance condition.

Unfortunately, whatever the approach and whatever its role, the theme is not provided with clear definitions. Besides, depending on the approach, the analysis of a given sentence will not give similar results, i.e. the theme will not correspond to the same element of the sentence.

Theme and rheme are two notions that cannot be separated. Yet in the literature their status is quite different. The theme is always considered as the relevant part of the sentence whereas the rheme

corresponds to the unrelevant part; the theme is the first element of the sentence, the rheme is the rest.
The rheme is always defined in terms of the theme.

2.2.1.1.3 PROBLEMS

The notion of theme is never conceptualized. Authors give examples or tests (question, negation, paraphrase) which are supposed to help the reader to recognize, in an intuitive way, what is thematic and what is not.

The following questions should be considered :

- Is this notion necessary in linguistics, and particularly in computational linguistics and text generation ? In what way ?

- Is there one notion, two notions ? It seems that the notion of theme has been used at several levels : sentence/utterance, text/discourse. At the sentential level the central issue is predication, at the textual level it is cohesion. So is it the same notion at both levels?

We will give our point of view in 2.2.2.1.

2.2.1.1.4 DEFINITIONS

The first linguists to show interest in the communicative issue belonged to the Prague Circle. They wanted to stress the functional aspect of communication and explain why in Czech or other Slavic languages a different word order in the sentence would convey a different meaning, or more precisely, different communicative effects. Linguists of the Prague Circle wrote a manifesto composed of nine theses published in the Travaux du Cercle Linguistique de Prague (TCLP, later TLP).

Mathesius tried to give a definition of the theme/rheme pair. In the second thesis he called this bipartition of the sentence 'the actual division of the sentence', which in English later became 'the Functional Sentence Perspective' (FSP).

Mathesius gave two definitions of the FSP. The first one in 1939 : "The starting point of the utterance is what is known or at least obvious in the given situation and from which the speaker proceeds and the core of the utterance is what the speaker states about, or in regard to, the starting point of the utterance". The second one in 1942 : "The foundation (or the theme) of the utterance is what is being spoken about in the sentence and the core is what the speaker says about this theme".

These two definitions reveal two basic aspects of the FSP : the contextual and the thematic (or structural). With the second definition Mathesius tried to consider the theme in purely linguistic terms, without referring to the cognitive or psychological aspects. There are good reasons to make this distinction since the two aspects do not always coincide : for instance a theme can convey an unknown information.

Mathesius' definitions gave rise to two main trends in the study of the communicative structure. One trend focused on the structural aspect and on the syntactic constructions, the other on the contextual aspect and on the theme as a cohesion factor. The first one was more concerned with the sentence level, the second one with the text level.

2.2.1.2 PRESENTATION OF THE NOTION

The notion of theme is a complex notion. [Escalier & Fournier, 1989] tried to give a clear account of the different approaches in the literature, finding out the various 'primitive' notions involved and the linguistic markers associated to these notions. We also tried to assign these notions a descriptive level : syntactic, semantic, communicative. But of course the borders are not so clear. For instance, word order, which is a typical syntactic phenomenon, appears at the three levels.

LEVELS	SUB-LEVELS	NOTIONS	DEVICES
Syntactic		Word order Typology of languages	Syntactic constructions
Semantic	Case & Lexical	Case weight Semantic content	Word order reflecting case distribution Lexical markers
Logical	Referential	Assertion vs presupposition Old/Given vs New	Lexemes Focalization Topicalization Deixis, Anaphora
Commun- icative	Cognitive	Topic Discourse aim	Topicalization
	Functional	Communicative dynamism (CD) Distribution of information	Word order reflecting basic distribution of CD

2.2.1.2.1 THE SYNTACTIC LEVEL

1 - TYPOLOGY AND WORD ORDER

Linguists and philosophers have extensively studied the problem of universals and typologies. We will not go into details here but just mention work related to our issue.

a - Six types of languages

[Greenberg, 1963] and other authors proposed a typology of languages with word order as a classifying criterion. There are three basic grammatical functions : subject, verb and object. Therefore there are six types of languages : SVO, SOV, OVS, OSV, VOS, VSO.

b - Sensitivity to FSP

The class to which a language belong is not the only factor. The fact that word order is more or less rigid is also important. Mathesius claimed that FSP is a dominant factor or a 'central' phenomenon in languages with free order, and that FSP is a 'peripheric' phenomenon in languages with a more rigid order.

c - Two orders

Several linguists classified word order into two classes.Greenberg made a distinction between the 'dominant order' (always possible but not always the most frequent) and the 'recessive order' (possible in certain conditions). Mathesius mentions an 'objective order' (when the theme is the left most constituent) and a 'subjective order' (when the speaker wants to start with the new information). Henri Weil, published in 1844 a book which greatly influenced Mathesius [Weil, 1844]; he mentions a 'pathetic order' when the order does not reflect the analytical character of thought. Pathetic order is used to convey emotive reactions. Jakobson talks about a 'neutral order' and an 'emotive order'. The marked order can be constrained by conditions which are not always grammatical.

In unmarked or neutral cases, when the 'natural' word order is respected, some authors consider the left-most constituent of the sentence as the theme. So, in declarative sentences, the theme will be a noun phrase, the grammatical subject ; in interrogatives the theme will be the interrogative word, and in imperatives, the imperative form of the verb [Brown & Yule, 1983].

Van Dijk [van Dijk, 1977] gave a list of semantic patterns reflecting the natural order of the elements (such as *general - particular, whole - part/component, set - element, including - included, large - small, outside - inside, possessor - possessed*). Unfortunately, he did not systematically study the effects of the reversal.

2 - SYNTACTIC CONSTRUCTIONS

Authors interested in syntactic constructions were most often generativists who worked on the English language. These authors were not interested in the semantic discrepancies produced by different word orders, as were the linguists from the Prague circle, but on the contrary they considered sentences with different word orders as semantically equivalent and described the various possible syntactic transformations preserving meaning. They compared marked sentences with unmarked ones, for instance, they considered that a passive sentence is derived from the active, that a topicalized sentence is a transformation of the non topicalized one.

What are the marked constructions? Can they be classified?
The syntactic constructions are described in heterogeneous terms, such as :
- syntactic categories (NP, VP, PP)
- grammatical functions (subject, complement)
- semantic cases (agent, dative).

The classifying criteria are not so clear, they seem to be :
- the position of the topicalized group (left/medial/final)
- the reprise or not of the topicalized element
- the absence or presence of preposition.

2.2.1.2.2 THE SEMANTIC AND LOGICAL LEVEL

1 - Case semantics

One of the basic hypotheses in the Prague school is that the linearity of the utterance reflects the order in which phenomena occur in extra-linguistic reality. The consequence is that a relative weight can be assigned to semantic cases according to the new status of information conveyed by lexemes corresponding to the various cases. For instance, 'in the world', the agent exists before the action he undertakes; the goal is achieved once the action has started. Therefore the informative weight of the agent will be inferior to that of the action, and the weight of the action inferior to that of the goal. In the sentence expressing the 'reality', there is a 'basic distribution' of semantic cases (agent < action < goal) which may correspond to that of the communicative dynamism or to the unmarked word order. It may correspond to it but it may also disturb it.

2 - Lexical semantics

Particular lexemes play a specific role in the thematic structure. They may have a value (thematic or rhematic) or an effect (thematizing or rhematizing) on other lexemes or phrases.

a - Verbs

Karel Pala [Pala, 1974] tried to classify verbs taking into account the semantic content of the predicate and its arguments as well as the relative weight of semantic cases. The semantic analysis helps to establish the various thematic structures of simple sentences, contextually independent. For instance, why say "*A girl came into the room*" ? According to the expected order (the one corresponding to the basic CD distribution), the known element, marked by the definite article ("into *the* room"), should be at the beginning of the sentence, whereas the unknown element, marked by the indefinite article ("*a* girl"), should be at the end. Why is the natural order reversed? The hypothesis is that the verb "*to come*" belongs to a class where the subject-verb order is respected but where the known-unknown order is reversed.

b - Determiners

To Firbas, referential determiners play a thematic role, or more precisely, they make thematic the NP they determine. On the contrary, indefinite articles will have a rhematizing effect. Several linguists studied this issue and found out the distinctions were not so clear. The role of determiners does not only depend on their class (definite or indefinite) but also on their value (e.g. generic or specific).

c - Other lexemes

A lexeme such as "even" is often mentioned in the literature as a typical rhematizing operator.

3 - Referential semantics

a - Given and new

Several authors, linguists and psycholinguists studied the given/new status of information [Halliday, 1967] [Brown & Yule, 1983] [Danes, 1974], [Iordanskaia, 1989] [Givon, 1983]. Prince [Prince, 1981] provides the basis for an extended taxonomy. We quote here [Brown &

Yule, 1983]. To Prince, a text is 'a set of instructions on how to construct a particular discourse model. The model will contain discourse entities, attributes and links between entities.

There are three kinds of entities in her taxonomy:

1 - New entities
- brand new: entities assumed not to be known in any way to the speaker
- unused: assumed by the speaker to be known to the hearer, in his background knowledge but not in his consciousness at the time of utterance.

2 - Inferrables
entities which the speaker assumes the hearer can infer from a discourse entity which has already been introduced.

3 - Evoked
- situationally: salient in the discourse context
- textually: already been introduced into the discourse which is now being referred to for second or subsequent time.

Brown and Yule added a further distinction in the category of textually evoked entities:
- current: introduced as 'new' immediately before the current new entity was introduced
- displaced: introduced prior to that.

Here are the categories of entity identified by Prince and refined by Brown and Yule with the forms of expression used to refer to them.

1 - New entities
a - brand new
(i) draw *a black triangle*
(ii) draw *a straight line*
(iii) there's *a circle* in the middle

b - unused

2 - Inferrable entities
(i) it's right through *the middle* (circle)
(ii) you start at *the edge* (triangle)
(iii) with *the right angle* (triangle)

3 - Evoked entities
a - situational
(i) in the middle of *the page*
(ii) *you've* got a triangle

b - textual-current
(i) to the left of the red line, about half a centimetre above *it*
(ii) there's a black circle...
above *it* there's

c - textual-displaced
(i) draw *a black triangle...*
underneath *the triangle*

Brand new entities are introduced by "*a*"; inferrables by definite expressions. Evoked situational forms are mostly used to mention the page which the hearer is drawing on. Most of the expressions used to mention current textual entities are either pronominal or elided, though there are some definite referring expressions. Displaced textual entities are never referred to pronominally or elided, but always referred to by a definite referring expression.

b - Assertion/Presupposition

We will not develop this aspect, traditionally studied by logicians, which is another way of considering the given/new status of information.

2.2.1.2.3 THE COMMUNICATIVE LEVEL

This level is concerned with the cognitive and functional aspects of communication.

1 - Cognitive aspects

There are several cognitive aspects studied by authors, mostly psychologists or psycholinguists:

- communicative intention of the speaker
- what is stored in memory and what is selected
- what is shared by both speaker and hearer.

To van Dijk [van Dijk, 1981] "taking a theory of cognitive information processing as one of the supporting theories for a theory of pragmatics, a cognitive account of the topic-comment distinction would be given in terms of mutual knowledge of speakers and hearers, intentions of speakers, and notion such as attention. New information can be processed only in relation to old information. This new information is tied to a concept, which in the present conversation should be foregrounded by the hearer, ie drawn from memory, and serve as a peg to hang on the new information. The pragmatic constraints on language use tell us that in principle the hearer is only interested in information he not yet has and that the information given must be relevant to the actual context."

2 - Functional aspects

Firbas refused the contextual and structural dichotomies known/new and theme/rheme. He tried to avoid a strict bipartition of the sentence, that is, the division of the sentence into two distinct parts, the theme and the rheme.

He proposed a gradual notion, the COMMUNICATIVE DYNAMISM, but unfortunately he did not give a continuous representation of this phenomenon. To the pair he added a third element, the transition, which most of the time is considered rhematic. Firbas also added three more notions: the theme proper (the element in the theme with the lowest CD degree), the rheme proper (the element in the rheme with the highest CD degree) and the transition proper (the element in the rhematic part with the lowest degree). These extra notions do not solve the problem; they do not represent the gradual aspect of communicative dynamism.

Yet communicative dynamism is an interesting notion : it is "the extent to which the sentence element contributes to the development of the communication". Instead of a bipartition the CD distributes various degrees to the elements of the sentence. The theme, which can convey a new information, will still be the element of the sentence with the lowest CD degree. The CD basic distribution is realized by word order in the sentence and gradually goes from less informative to more informative. This distribution is supposed to reflect the character of human thought and the linear aspect of communication.

2.2.2 PROPOSAL OF A METHOD

2.2.2.1 OUR APPROACH AND TERMINOLOGY

We have presented in part 2.2.1 the notions of theme and rheme and the use of these notions as a means of analyzing the communicative function of the sentence. Our idea is that the distinction theme/rheme is necessary in linguistics and in computational linguistics, and more precisely in analysis, but it is not a sufficient tool. The theme/rheme analysis may give us indications on the thematic/rhematic progression in the text. We can see how themes and rhemes interrelate from one sentence to the next and see how they relate to the topics. The theme/rheme distinction, though at the sentential level, probably gives more information on the text than on the sentence itself.

We wish to make a distinction between the pair topic/comment and the pair theme/rheme. This distinction is not related to the difference between a textual and a sentential level but to the difference between a conceptual level and a linguistic one, that is, between a conceptual content and its linguistic expression.

As for the labeling of the main phenomenon we are concerned with, there are at least two possible pairs: thematic vs communicative, structure vs organization.

THEMATIC focuses too much on the 'theme' element of the sentence and on the 'topicalization' devices.

COMMUNICATIVE seems better in the sense that it focuses on the effect or goal wanted by the author or the effect produced by the utterance. This kind of effect is often considered 'communicative'.

The opposition STRUCTURE vs ORGANIZATION reflects two different perspectives : static vs dynamic. It would probably be more coherent with the approach chosen for the extraction of topics to consider the communicative phenomenon as a dynamic process, as an organization procedure. Besides, even if we are now in a descriptive phase, our final aim is generation. Therefore it makes sense to see a topic as an extraction procedure and communicative organization as the dynamic use of linguistic devices constrained by the desired goal and expressing it. STRUCTURE would be too close to syntax whereas ORGANIZATION could include other devices beside syntactic ones. So we favour the label 'COMMUNICATIVE ORGANIZATION'.

We will study four aspects:

- Communicative organization
- Distribution of information
- Text cohesion
- Thematic progression.

For each of these aspects, we will describe:

- The existing linguistic devices
- The corpus analysis
- The problems and results.

We will also examine the relation between the topic-comment structure and the theme-rheme structure.

2.2.2.2 - COMMUNICATIVE ORGANIZATION

2.2.2.2.1 - LINGUISTIC DEVICES

Here is a list of linguistic devices considered as a means to convey communicative effects. Once the linguistic devices are listed it would be nice to have the corresponding effects. Unfortunately the effects (such as emphasis or contrast) are notions which are largely mentioned in the literature but still quite vague. These devices are gathered here but in fact they have been accounted for up to now by various fields such as rhetorics, stylistics, argumentation... Our idea here is to consider that all these phenomena should be tackled from a unique point of view and included into the list of devices at hand for organizing the sentence.

- Lexical selection
  - Converses
- Determiners
- Syntactic constructions
  - Topicalisation
  - Cleft and pseudo-cleft sentences
  - Active vs passive
  - Nominalisation
  - Tagging and reprise
  - Movements (dative, adverbs, ...)
  - Complex sentences
  - ...

2.2.2.2.2 CORPUS ANALYSIS

1 - CONVERSES

With Banys [Banys, 1984] we consider that the choice of a certain lexeme among all the converses (e.g. BUY vs SELL vs COST) is a means of expressing specific communicative effects. But how do we know what lexemes are converses? This kind of information is not given by usual dictionaries. We listed all the verbs/predicates in the text with their participants/arguments and then we compared the role/position of the arguments. If the semantics of the verbs is the same but if the role of the participants, we can consider them as converses.
See Appendix B1 for the list of predicates and arguments.

2 - DETERMINERS

We mention determiners here since they are considered as playing an important part in the communicative organization. But we will study them more closely in relation to
- the status of information (given vs new) and the distribution of information in the sentence (from given to new) ;
- text cohesion (ensured by referential links) .
See Appendix B2 for the list of noun phrases with their determiners.

3 - SYNTACTIC CONSTRUCTIONS

Here are examples from Chet Creider's paper 'On the explanation of transformations'. There are two kinds of rules, the topicalizing rules and the focusing rules : the first ones concern movements to the left, the second ones movements to the right. This list is not an exhaustive one but it will give an idea of the number of devices offered by English. We will then see whether these various devices are used in our corpus.

a - Topicalizing rules

- Topicalization
  (1) *I can eat English muffins every morning.*
  (1') *English muffins I can eat every morning.*
- Left-dislocation
  (2) *I hope to meet Griselda's husband someday.*
  (2') *Griselda, I hope to meet her husband someday.*
- Passive
  (3) *John did the artwork.*
  (3') *The artwork was done by John.*
- Dative Movement
  (4) *I gave the book to George.*
  (4') *I gave George the book.*
- about-Movement
  (5) *Mord talked to the Njalssons about Hoskuld.*
  (5') *Mord talked about Hoskuld to the Njalssons.*
- Adverb Fronting
  (6) *I hope to return home the day after tomorrow.*
  (6') *The day after tomorrow I hope to return home.*
- Particle Movement
  (7) *He wore out the valve.*
  (7') *He wore the valve out.*
- Subject Raising
  (8) *That the interface will go down while we are on line is virtually certain.*
  (8') *The interface is virtually certain to go down while we are on line.*

b - Focusing rules

- Extraposition (it-Insertion)
  (9) *That the interface will go down while we are on line is virtually certain.*
  (9') *It is virtually certain that the interface will go down while we are on line.*
- there-Insertion
  (10) *An Irish Rover is in the garden.*
  (10') *There is an Irish Rover in the garden.*

- Extraposition from NP
  (11) *The man who won was praised by the press.*
  (11') *The man was praised by the press who won.*
- Complex NP Shift
  (12) *I consider the problem of keeping the house warm in the winter unsolvable.*
  (12') *I consider unsolvable the problem of keeping the house*

the winter.

- Quantifier Postposing

(13) *All the linguists in this room know at least one language.*

(13') *The linguists in this room all know at least one language.*

See Appendice B3 for the syntactic analysis of the corpus.

2.2.2.2.3 PROBLEMS AND RESULTS

1 - CONVERSES

We first had a look at the list of predicates without considering their arguments and extracted some possible converses. In each column synonyms are separated by a slash.

release/send/transfer/transmit	*accept/receive/come from*
lead	*come from*
generate/produce	*result*
require	*enable*

We then looked at these possible converses with their arguments.
- It seems that *send*, *transfer* and *transmit* concern data (value, failure). The argument of *send* is either a failure or a request. Can a request be assimilated to a command?

- With *accept, receive, come from* and *release* we have a more homogeneous class concerning commands. So here we could consider *accept, receive* as synonyms and *release* as their converse. Let us examine *release* : it is a three-argument predicate with agent, object, recipient. When the focus is on the agent, we have *release*, when it is on the object, we have *come from* (a command comes from the agent), and when it is on the recipient, we have *accept* or *receive*.

- *produce* is very specific, it only concerns power.
- *result* is result(command,action).
So they are not converses.

- Looking at the arguments it seems that *enable* and *require* are not converses.

The list of converses, without looking at the types of arguments, is already restricted; all the more when taking into account the arguments. The quasi absence of converses can be explained by the extensive use of passive forms, quite usual in technical texts.

2 - SYNTACTIC CONSTRUCTIONS

Here are some conclusions concerning syntactic constructions conveying particular communicative effects. This is the number of occurrences in parts 1 & 2 but parts 4 and 5 have been examined too and confirm these results.

- Active vs passive	147 vs 224
- Topicalizations	0
- Cleft sentence	1
- Inversion verb-subject (after an initial PP)	1

- Impersonal form		1
- Relative clauses		6
- Completives		4
- Infinitives		5
- Prepositional phrases	(modifiers)	13
Initial		3
Final		4
- Adverbials mostly textual, spatial or temporal		40
Initial		11
Final		9
- Nominalizations (among them 20 are derived from verbs present in the text).		33

- The most noticeable phenomenon is the anteposition of the subordinate clauses : 19 (+ 4 if we include complex ones) compared to 7 (+ 2) for the reverse order.

Compared to Creider's list, the available syntactic constructions are poorly used. The communicative organization is quite simple and regular as far as lexical devices (converses) and syntactic constructions are concerned.

The main phenomena are

- the number of nominalizations,
- the anteposition of subordinate clauses and
- the extensive use of passives.

2.2.2.3 DISTRIBUTION OF INFORMATION

2.2.2.3.1 LINGUISTIC DEVICES

We checked whether the 'natural' order of the arguments is respected and tried to determine the values of the determiners : they give indications about the new/old status of information.

The idea is that the natural order of the various elements in the sentence goes from known to unknown.

2.2.2.3.2 CORPUS ANALYSIS

1 - ORDER OF ACTANTS

Many linguists, interested in universals and the typology of languages, claim that the order of the arguments is meaningful for several reasons, linguistic and cognitive. It has to do with the status of information and the cognitive weight of the arguments. Once the order of the actants is described we can see in the text if this order is always the same or not.

See Appendix B1 for the list of predicates and arguments.

2 - DETERMINERS

We had a look at the different meanings or values of the determiners in the text and identified the part they play in the communicative organisation, especially regarding the information status and anaphora.
See Appendix B2 for the list of determiners with their values.

2.2.2.3.3 PROBLEMS AND RESULTS

As a general conclusion, the distribution of information from known to unkown in the sentence is very regular.

1 - ORDER OF ACTANTS

One of the main problems here is the distinction between modifiers and arguments. Modifiers are generally considered as syntactically and semantically optional whereas arguments may in some cases be omitted but are semantically part of the meaning of the predicate. Unfortunately the distinction is not easy to make, though it would be useful, e.g. in case of anteposition which is a more significative phenomenon when dealing with an argument rather than a modifier (the initial position of the latter remains to be evaluated). The other problem is the existence of a 'canonical' or 'natural' order of arguments in a predicate. To Sells (1985) "many different theories make reference to theta-roles (under one name or another) yet there is unfortunately no presently available theory of what the range of possible roles is and how you might tell in a given context which one one you are dealing with; one must, for the present, rely on intuition in a large part".

The other question concerns cases: are they hierarchalized Fillmore (1968) (1969) proposed the following case hierarchy:
AGENT>EXPERIENCER>INSTRUMENT>OBJECT>SOURCE> GOAL>LOCATION>TIME. In Jackendoff (1972) we find another hierarchy: THEME>GOAL>SOURCE>BENEFICIARY.

As a subject selection principle (not as an absolute hierarchy of arguments), Fillmore's hierarchy seems to work. Thus the order of actants seems quite regular except for:
- passives (the extensive use of agentless passives follows Fillmore's hierarchy);
- anteposition of adverbials and prepositional phrases

- Location
- Time
- Textual.

These exceptions are not surprising in such a corpus.

2 - DETERMINERS

The idea is to check whether noun phrases at the beginning of a sentence are definite and convey given information while noun phrases at the end of the same sentence are indefinite and convey new information. So we had a look at the various determiners and their values in each sentence.

There are several patterns within one sentence:

- only *the*
- only *a*
- no determiner
- progression from *the* to *a* or numerals
- *the* can be at the end when it introduces a new unique item.

There are few exceptions:

- passives with agent
- mistakes: for instance an *a* at the end of the sentence is not the indefinite article accompanying an unknown item but is a possessive or a distributive marker.

2.2.2.4 COHESION

2.2.2.4.1 LINGUISTIC DEVICES

Halliday and Hasan tried to determine how a set of sentences does or does not constitute a text. A set of sentences will constitute a text if the sentences are linked together by 'cohesive' relationships.
Halliday and Hasan listed the various types of cohesive devices in English. Thus they listed the linguistic resources available to a speaker or writer to mark cohesion in a text.

Here are the various types of cohesion as listed by Halliday and Hasan.

A - REFERENCE

I - Pronominals	
1 - singular, masculine	*he, him, his*
2 - singular, feminine	*she, her, hers*
3 - singular, neuter	*it, its*
4 - plural	*they, them, their, theirs*

II - Demonstratives and definite article	
1 - demonstrative, near	*this/these, here*
2 - demonstrative, far	*that/those, there, then*
3 - definite article	*the*

III - Comparatives	
1 - identity	*same, identical*
2 - similarity	*similar(ly), such*
3 - difference	*different, other, else, additional*
4 - comparison, quantity	*more, less, as many,* ordinals
5 - comparison, quality	*as* + adjective, comparatives and superlatives

B - SUBSTITUTION

I - Nominal substitutes	
1 - for noun head	*one/ones*
2 - for nominal complement	*the same*
3 - for attribute	*so*

II - Verbal substitutes	
1 - for verb	*do, be, have*
2 - for process	*do the same/likewise*
3 - for proposition	*do so, be so*
4 - verbal reference	*do it/that, be it/that*

III - Clausal substitutes	
1 - positive	*so*
2 - negative	*not*

C - ELLIPSIS

I - Nominal ellipsis
1 - Deictic as head
- specific deictic
- non-specific deictic
- post-deictic

2 - Numerative as head
- ordinal
- cardinal
- indefinite

3 - Epithet as head
- superlative
- comparative
- others

II - Verbal ellipsis
1 - lexical ellipsis
- total
- partial

2 - operator ellipsis
- total
- partial

III - Clausal ellipsis
1 - propositional ellipsis
2 - modal ellipsis
3 - general ellipsis of the clause
4 - zero (entire clause omitted)

D - CONJUNCTION

I - Additive	
1 - simple	
- additive	*and, and also*
- negative	*nor, and... not*
- alternative	*or , or else*
2 - complex, emphatic	
- additive	*furthermore, add to that*
- alternative	*alternatively*
3 - complex, de-emphatic	*by the way, incidentally*
4 - apposition	
- expository	*that is, in other words*
- exemplificatory	*eg, thus*
5 - comparison	
- similar	*likewise, in the same way*
- dissimilar	*on the other hand, by contrast*
II - Adversative	
1 - adversative proper	
- simple	*yet, though, only*
- + and	*but*
- emphatic	*however, even so, all the same*
2 - contrastive (avowal)	*in (point of) fact, actually*
3 - contrastive	
- simple	*but, and*
- emphatic	*however, conversely, on the other hand*
4 - correction	
- of meaning	*instead, on the contrary, rather*
- of wording	*at least, I mean, or rather*

5 - dismissal	
- closed	*in any/either case*
- open-ended	*in any case, anyhow*
III - causal	
1 - general	
- simple	*so, then, therefore*
- emphatic	*consequently*
2 - specific	
- reason	*on account of this*
- result	*in consequence*
- purpose	*with this in mind*
3 - reversed causal	*for, because*
4 - causal, specific	
- reason	*it follows*
- result	*arising out of this*
- purpose	*to this end*
5 - conditional	
- simple	*then*
- emphatic	*in that case, in such an event*
- generalized	*under the circumstances*
- reversed polarity	*otherwise, under other circumstances*
6 - respective	
- direct	*in this respect, here*
- reversed polarity	*otherwise, apart from this, in other respects*
IV - Temporal	
1 - simple	
- sequential	*then, next*
- simultaneous	*just then*
- preceding	*before that, hitherto*
2 - conclusive	*in the end*
3 - correlatives	
- sequential	*first... then*
- conclusive	*at first/originally/formerly... finally/now*
4 - complex	
- immediate	*at once*
- interrupted	*soon*
- repetitive	*next time*
- specific	*next day*
- durative	*meanwhile*
- terminal	*until then*
- punctiliar	*at this moment*
5 - internal temporal	
- sequential	*then, next*
- conclusive	*finally, in conclusion*
6 - correlatives	
- sequential	*first... next*
- conclusive	*in the first place... to conclude with*
7 - here and now	
- past	*up to now*
- present	*at this point*
- future	*from now on*
8 - summary	
- summarizing	*to sum up*
- resumptive	*to resume*
V - Other (continuative)	*now, of course, well, anyway , surely, after all*

E - LEXICAL

I - Reiteration
1 - Repetition
2 - Synonym or near-synonym
3 - Superordinate
4 - General item

II - Collocation
1 - Opposites
- complementaries
- antonyms
- converses
2 - Ordered series
3 - Semantic field

2.2.2.4.2 CORPUS ANALYSIS

The list of determiners in Appendix B2 gives indications on the repetition of a given NP within a sentence or a paragraph. These indications, of course, are much more detailed than the results of the analysis of the TP. As for the other types of cohesion listed by Halliday & Hasan, see Appendix B4.

2.2.2.4.3 RESULTS

REFERENCE 150			
1- Pronominals			11
	it	4	
	they	1	
	its	5	
	their	1	
2 - Demonstratives and definite article			120
1 - near	*this, these*	8	
3 - definite article	*the*	111	
3 - Comparatives			15
1 - identity	*same*	3	
3 - difference	*different*	1	
	other	2	
4 - quantity			
ordinals	*first*	1	
5 - quality			
comparatives	*higher*	2	
	greater	1	
superlatives			
	highest	1	
6 - others			
	too	1	
	also	1	
	respectively	2	

SUBSTITUTION			7
1 - Nominal substitutes			
	one	3	
2 - Verbal substitutes			
	do	4	
ELLIPSIS			18
1 - Nominal ellipsis			
	the first	1	
	the highest prioritied	1	
2 - Verbal ellipsis			
	be	9	
	will	2	
	can	2	
3 - Clausal ellipsis			
preposition			
	to	3	
CONJUNCTION			16
1 - Additive			
- complex			
	furthermore	2	
- apposition			
expository	*that is*	1	
	in other words	1	
exemplificatory	*eg*	1	
- comparison	*in the same way*	2	
3 - Causal			
- general			
	so (that)	2	
- reversed causal			
	because	1	
- conditional			
	in this case	2	
4 - Temporal			
	then	2	
	at the same time	2	

LEXICAL COHESION (only nouns)

1 - Reiteration
- 1 - Repetition
  - *DG*
  - *switch*
  - *start request*
  - *running input*
  - *start blocking output*

2 - Synonym or near-synonym
- *standby sequence*
  - *master/standby sequence*
  - *priority sequence*
- *control modes*
  - *control possibilities*
  - *modes of operation*
- *selector*
  - *switch*
- *consumer*
  - *power consumer*
- *generating set*
  - *generator*

3 - Hyperonym
- *GS*
- *function*
- *action*
- *condition*
- *mode*
- *sequence*
- *signal*

4 - General word
- *situation*
- *case*

2 - Collocation

1 - Opposites
- Antonyms
  - *connection/disconnection*
  - *start/stop*
- Converses
  - *generator/consumer*

2 - Ordered series
- *start sequence*
- *stop sequence*
- *priority sequence (the next/the former/the first)*

3 - Semantic field
- Parts/Whole
  - PMS and its components
- Verb/Nominalization

As far as reference is concerned, the cohesion is not so strong. The figures are quite misleading. Determiners in English are difficult to handle for foreigners.

We noticed:
- the repetition of the NP within the same sentence instead of the use of a referential term (causes doubt in the mind of the reader);
- the bad choice of referential terms
- the extensive use of the zero-determiner (not only in headings)
- the extensive use of compounds.

See list of suggestions in Appendix B5.

In Halliday and Hasan's sample texts, the number of occurrences of each cohesive device is the following :

- Lexicon	107
- Reference	81
- Conjunction	31
- Ellipsis	26
- Substitution	10

Here the order is nearly the same. Cohesion mostly comes from lexical cohesion. This is not surprising in a technical document dealing with a restricted and specific domain. Apart from the fact that most terms are technical and belong to the same domain, the cohesion is also ensured by the presence of names denoting components, actions performed by these components, and data transmitted by one component to another.

2.2.2.5 THEMATIC PROGRESSION

2.2.2.5.1 LINGUISTIC DEVICES

Here are some elements on the thematic progression as approached by Danes in his paper 'Functional Sentence Perspective and the organization of the text'.

'Thematic progression' means to Danes 'the choice and ordering of utterance themes, their mutual concatenation and hierarchy, as well as their relationship to the hyperthemes of the superior text units (such as the paragraph, chapter, ...), to the whole text, and to the situation. Thematic progression might be viewed as the skeleton of the plot.'

1 - Method

Danes uses the wh-question as a criterion for detecting the theme and the rheme of a given utterance. The part common to the question and the answer will correspond to the theme and the answer to the question, i.e. the new information, will correspond to the rheme.

A complex sentence is reduced to a series of sentences with a simple T-R structure.

For instance, the sentence

"Wohler heated ammonium cyanate and found that it was thereby converted into urea, previously known only as a product of living organisms."

is reduced to

(a) *Wohler heated some ammonium cyanate.*
(b) *He found that it was thereby converted into urea.*
(c) *This substance had been previously known only as a product of living organisms.*

In the complex sentence:
(b) has lost its independent status and has been combined with (a)
(c) has lost its independent status and its structure T-R as well.
It has been restricted to its rhematic elements and fused with Rb into a complex R(b,c).

So the FSP structure of the sentence is Ta -> Ra + Tb (= Ta) -> R(b,c).

2 - Kinds of utterances

There are three kinds of utterances :

- simple utterances

- composed utterances (composition)
  - multiple utterances :
    *"Goethe wrote the second part of Faust after eighty,
    and Hugo astounded the world with Torquemada at eigthy."*
    T1 -> R1 and T2 -> R2

  - utterance with a multiple T :
    *"The melting of solid ice and the formation from ice of
    liquid water exemplify physical changes."*
    T1 and T2 -> R

  - utterance with a multiple R :
    *"It is further postulated that the activated amino acids
    are joined together... and that the long chains are molded
    in a specific manner..."*
    T -> R1 and R2

- condensed utterances (fusion)

  - utterance with a complex T :
    *"This dark-coloured liquid, known as crude oil, is obtained
    from wells of different depth."*
    T1 = T2
    T2 is deleted
    R1 is fused with T1 (R1 is thematized)
    R2 becomes the R of complex T

  - utterance with a complex R :
    "The amino acids are required for making proteins, consisting of long chains of these units."
    T2 = R1
    T2 is deleted
    R2 is fused with R1 (R2 is rhematized)
    R2 + R1 becomes a complex R

3 - Comparison of fusion and composition

Composition with a multiple R and fusion with a complex R are not equivalent.

1 - Composition:		A -> B
	A -> C	
Result:	A -> B and C	
2 - Fusion:		A -> B
	B -> C	
Result:	A -> B + C	

In the first case, B and C are both related to A. In the second, C is related to B. In the first case, the second occurrence of A is deleted, in the second case, the second occurrence of B is deleted.

Syntactically, there are differences:

1 - coordination: "and", ","
2 - relative clause or present participle in the rheme.

Composition with a multiple R and a fusion with a complex T are closer but they are not equivalent either.

In both cases:

A -> B
A -> C

but, in the second case, A being deleted, C is thematized, whereas in the first case it is still a rheme.

Syntactically:
1 - coordination
2 - relative or apposition in the theme.

4 - Types of thematic progression

There are three main types of thematic progression

1 - Simple linear TP (linear thematisation of rhemes)

```
T1 -> R1
      |
      T2 (=R1) -> R2
                  |
                  T3 (=R2) -> R3
```

Each R becomes the T of the next utterance.

Ex: *The first of the antibiotics* [T1] *was discovered by A.F. in 1928* [R1]. *He* [T2] *was busy at the time..* [R2].

```
T1 -> R1
      |
      T2 (=R1) -> R2
```

2 - TP with a constant theme

```
T1 -> R1
|
T1 -> R2
|
T1 -> R3
```

3 - TP with derived Ts

```
                [T]
T1 -> R1
          T2 -> R2
                    T3 -> R3
```

The themes are derived from a hypertheme.

There are various possible ways of combining the main types.

1 - Split rheme

```
T1 -> R1 (= R'1 + R"1)
            |    |
            |    |
 ---------------  |
|                 |
T'2 -> R'2              |
                  |
 -----------------------
|
T"2 -> R"2
```

Example: "*All substances can be divided into two classes: elementary substances and compounds. An elementary substance is ... A compound is ...*"

2.2.2.5.2 CORPUS ANALYSIS

In Appendix B6 we give the analysis of the thematic progression in the corpus. We describe each sentence with its bipartition theme/rheme; we then give the thematic progression of whole chapters in order to show the various types of thematic progression.

2.2.2.5.3 PROBLEMS AND RESULTS

The various problems encountered during analysis are the following ones:

- Is the theme the left-most constituent, the grammatical subject or the part corresponding to the implicit wh-question? We favoured the grammatical subject but sometimes it raises a problem.

- Subordinate clauses can be considered as:
    - belonging to the rheme
    - belonging to the theme when they are initial (otherwise the rheme would contain two distinct parts)
    - having its own theme-rheme structure. The whole sentence can be seen as a composition.

- Anteposition of adverbials and appositions:
    - If the theme is the left-most constituent, then the adverbial or the apposition is the theme and both subject and predicate belong to the rheme;
    - If the theme is the grammatical subject, then the adverbial or the apposition may belong to the rheme or it neither belongs to the theme nor to the rheme.

- Level of decomposition
    - Do we consider noun phrases and verb phrases without decomposing further? Or do we extract relatives, prepositional phrases?
    - Do we claim a partial or total identity?

We generally did not decompose further than noun or verb phrases and did not claim a total identity.

The patterns are complicated but quite regular. All the patterns mentioned by Danes are present in the text; but others, not mentioned by Danes, were also realized, such as the constant rheme or the split theme. The order of importance is the following:

- Constant theme
- Linear TP
- Split rheme
- Constant rheme
- Split theme
- Hyperthemes

2.2.2.6 TOPICS AND THEMES

2.2.2.6.1 ANALYSIS

Here we will study the relation between the topic-comment structure and the theme-rheme structure. The idea is to find out whether there are some regularities and whether it will possible to have general rules expressing the correspondence between the two structures and allowing the transition from one structure to another.
See Appendix B7 for the comparison of topics and themes.

2.2.2.6.2 PROBLEMS AND RESULTS

In most cases:

- The first argument of the topic corresponds to the theme;
- The predicate of the topic is expressed by a verb or predicate;
- The comment corresponds to the rheme.

This triplet is quite close to what Firbas and Iordanskaja found out as the three relevant elements.

These conclusions help to check the stability of the topics, i.e. the type of comment they get as result, the types of the arguments and the order of arguments.

There are some questions yet unanswered:

- Should the theme be always present as argument of the topic?
- In case of several arguments in the topic:
  - Are they all expressed in the theme, that is, composed into a single NP;
  - Are they only means of extracting information and thus they should not be expressed?
  - Are they rhematic?

2.2.3 PROPOSAL OF A MODEL FOR GENERATION

We will very shortly suggest another approach which seems to us more adequate if we want to generate text with a good communicative organization.

We tried to elaborate a new model taking into account four main phenomena:

- Orientation
- Hierarchalization
- Topicalization
- Focalization.

Orientation concerns the 'valence', that is the predicate and its arguments. Depending on where the focus is, a different lexeme will be produced. This lexical selection is closely related to the choice between converses (*buy* vs *sell* vs *spend* vs *cost*) or between a verb and its nominalization or gerund (*sell* vs *sale*, *buy* vs *buying*).

Hierarchalization is a way of indicating:

- The main verb of the sentence
- He finite verbs of the sentence, allowing that way the choice between e.g. an epithet and a relative (*a blue book* vs *a book which is blue*).

Topicalization and focalization take in charge syntactic constructions such as the ones we listed.

2.2.4 REFERENCES

Banys, W. (1984) "Semantique, structure theme-rheme, syntaxe et lexique", in *Cahiers de Lexicologie*, 45, 2.

Benes, E. (1968) "On Two Aspects of FSP", *Travaux Linguistiques de Prague*, 3.

Boons, Guillet-Leclere (1976) *Constructions intransitives*, Droz.

Brown, G. & Yule, G. (1983) *Discourse analysis*, Cambridge University Press.

Cadiot, P. & Fradin, B. (1988)"Le theme en perspective", *Langue francaise*, 78.

Creider, C.A. (1979) "On the explanation of transformations", in T. Givon (ed.)*Syntax and semantics*, 12.

Dahl, O. (1969) "Topic and Comment: A study in Russian and General Transformational Grammar", in *Slavica Gothoburgensia*, Goteborg.

Dahl, O. (1974) "Topic-comment structure in a generative grammar with a semantic base", in F.Danes (ed.) *Papers on Functional Sentence Perspective, Prague.*

Danes, F. (1960) "Sentence intonation from a functional point of view", *Word*, XVI.

Danes, F. (1964) "A Three Level Approach to Syntax", *Travaux Linguistiques de Prague*, I, Prague.

Danes, F. (1967) "Order of Elements and Sentence Intonation" in *To honor Roman Jakobson*, The Hague.

Danes, F. (1968) "Some Thoughts on the Semantic Structure", *Lingua*, 21, 1968.

Danes, F. (1974) "FSP and the Organization of Text", in F.Danes (ed.) *Papers on Functional Sentence Perspective*, Prague, Academia.

Escalier, M.-C. & Fournier, C. (1989) "La structure theme-rheme", Internal Report, Dassault-Aviation.

Fillmore, C. (1968) "The Case for Case" in Bach and Harms (eds) *Universals in Linguistic Theory*, Halt, New York, Rinehart & Winston.

Fillmore, C. "Toward a Modern Theory of Case", in Reibel & Schane (eds) *Modern Studies in English-Readings in Transformational Grammar*, New Jersey, Prentice Hall.

Firbas, J. (1964) "On Defining the Theme in Functional Sentence Analysis"in *Travaux Linguistiques de Prague* I.

Firbas, J. (1965) "A Note on Transition Proper in Functional Sentence Analysis" in *Philologica Pragensia*, VIII, Prague.

Firbas, J. (1974) "Some Aspects of the Czechoslovak Approach to Problems of FSP", in F.Danes (ed.) *Papers on Functional Sentence Perspective*, Prague, Academia.

Givon, T. (1983) "Topic Continuity in Discourse", in (ed.) T.Givon, *Topic Continuity in Discourse*, Vol.3.

Greenberg J.H. (1963) "Some Universals of Grammar", in *Universals of Language*, Cambridge, Mass.

Haiman, J. (1978) "Conditionals are topics", in *Language*, 54,3.

Halliday, M.A.K. (1967-68) "Transitivity and theme in English", *Journal of Linguistics*.

Halliday, M.A.K. (1970) "Language Structure and Language function", in J.Lyons (ed.) *New Horizons in Linguistics*, Penguin.

Halliday, M.A.K. (1974) "The place of FSP in the system of linguistic description", in F.Danes (ed.) *Papers on Functional Sentence Perspective*, Prague, Academia.

Halliday, M.A.K. & Hasan, R. (1976) *Cohesion in English*, London, Longman.

Iordanskaja, L. (1989) "Communicative structure and its use during text generation", Internal Report, ORA, Montreal.

Jackendoff, R (1972) *Semantic Interpretation in Generative Grammar*, Cambridge, Mass., MIT Press.

Jakobson R. (1963) "Implications of Language Universals for Linguistics", in *Universals of Language*, Cambridge, Mass.

Paduceva, E.V. (1979) "Interaction de la structure communicative et de la structure des relations associatives", in *T.A.Informations*, 2.

Pala, K. (1974) "Semantic classes of verbs and FSP", in F.Danes (ed.) *Papers on Functional Sentence Perspective.*

Prince, E.F. (1981) "Toward a taxonomy of given-new information", in P.Cole (ed.) *Radical Pragmatics*, New York, Academic Press.

Sells, P. (1985) *Lectures on Contemporary Syntactic Theories*, Stanford, CSLI Lecture Notes 3.

van Dijk, T.A. (1977) *Text and Context*, London, Longman.

van Dijk, T.A. (1981) *Studies in the Pragmatics of Discourse*, The Hague, Mouton.

Weil, H. (1844) *De l'ordre des mots dans les langues anciennes comparees aux langues modernes*, Paris.

2.3 SYNTACTIC ANALYSIS

2.3.1 - Introduction

The text analysed here is :

SDS for ISC, Danyard NB 702.704

Power Management System

Pages 5.40-2 / 40.13

Pages 5.40-33 / 40.47

The goals of the analysis are :

(a) Specify the characteristic linguistic features of the text in order to generalise them ; these specifications are intended to be a crucial conditioning element for the generation of texts of the same class as the one analysed here.

(b) Define the major issues underlying an analysis of texts allowing to obtain (a).

(c) Define a general methodology of this type of text analysis.

Results in (a) are important keys for the evaluation of generators and of generators models in terms of their capacity to account for the generalised specifications.

The aims of the syntactic analysis are twofold :

(i) produce an analytical and compact description of the text units ; (ii) building on (i), specify salient and generalisable syntactic phenomena.

Section 2.3.2 deals with (i), and section 2.3.3 with (ii). Results in (i) are stored in a Text Data Base, while the ones in (ii) are organised in the Text Structures Knowledge Base.

2.3.2 Text units descriptions

The analytical table of Appendix C2 presents a macro-description of each of the text units of the analyzed text (sections 1 & 2). The results are summed up in the table at the end of this section.

The macro-description is intended to capture in a compact way the major indices revealing the verbal complexity of a singular text unit. In the column *Type phrase*, the top level syntactic structure of a particular unit is entered. The values of this column are :

(a) Co-s

(b) Sub-S1

Sub-S2

Sub-S3

Sub-S4

Sub-S5

(c) S

(d) cS+-

cS-+

(e) T

T:

St

Sub-St

Co-S covers coordinated top-level sentences. Ex :

1.1.2 : *The SG is connected to the Main Engine (ME) and it can produce power to either the busbar or the Bow- / Stern- Thruster (BT, ST).*

Some ad hoc choices have been made concerning the conjunction (see below 2.3.3.5).

The **labels in (b)** cover subordinate structures. See in 2.3.3.1 their values and examples.

S stands for simple sentences, i.e sentences with no top level coordinate structures, with no subordination and with neither embeddings nor coordination of verb forms.

cS stands for sentences with either some kind of embeddings or with coordination of verbal forms (see 2.3.3.4).

The **labels in (e)** cover nominal expressions. See in 2.3.3.2 their values and examples.

The next field of the table captures qualitatively and quantitatively the **verbal forms** present in a particular unit. A typology of six different forms has been used:

V1 : V [infl(ected), pres(ent)] ; full verbs including *be* and *have*.

V2 : be [infl, pres] + V [pass] ; passive constructions.

V3 : M [infl, pres] + V[base] ; M $\in$ {can, must, will}

modal constructions.

V4 : M [infl, pres] + be + V [pass] ; modal with passive construction.

V5 : Infinitives

V6 : Other than V1 to V5 :

have [infl, pres] + V[ed] ; perfective construction

be [infl, pres] + V [ing] ; progressive construction

do [infl, pres]

could + V[base]

The field *coordination* informs quantitatively and qualitatively about coordinate structures in the unit. See in 2.3.3.5 for the explanation of the labels C1 to C9.

The last two columns give quantitative results about "/" and about Embedding (= E). The different kinds of embeddings are presented in 2.3.3.3.

Summary table

Text	N														
secti ons	U ni t s	C o - S	S u b - S	S	c S	N e	V F	V 1	V 2	V 3	V 4	V 5	V 6	C	/	E
1 2.1	1 0 4 2	2 1	– 1	5 1 1	– 1	3 2 8	1 0 2 4	7 1 1	2 8	1 4	– –	– –	– 1	1 0 1 8	6 5	– 1
2.2 2.3 2.4 2.5	2 1 3 2 6 6 1 2	– 2 2 1	5 1 0 1 6 4	4 1 4 2 9 5	6 2 9 1	6 4 1 0 1	3 4 5 1 9 0 1 9	9 1 5 2 9 1 0	8 2 2 3 7 6	1 4 1 1 2	4 8 3 –	8 1 9 –	4 1 1 1	5 2 0 2 0 1	1 7 2 3 3	7 4 5 1
Total s	1 8 3	8	3 6	6 8	1 9	5 2	2 2 8	8 1	8 3	2 3	1 5	1 8	8	7 4	4 5	1 8

From the above table, it is possible to obtain relevant information on the adequacy (quantitative aspect) of texts submitted to analysis (see 2.7.2 (ii) and the 'Representativeness' checker in 2.7.3 Fig. 1). The table of Appendix 2 illustrates the type of information that a Text DB is intended to store (see 2.7.2 (vi), and the output of the text Analyzer component in 2.7.3).

2.3.3 Salient syntactic phenomena

The following sections 2.3.3.1 to 2.3.3.5 intend to evaluate quantitatively and qualitatively the most important syntactic phenomena of the text. The linguistic constructs are presented in these sections. See in the corresponding Appendices C3 to C7 the *Summary* and the *Distribution of occurrences* of each construct. Information of this kind, which is relevant to the Planner specification, is intended to be stored in the Text Structures KB (see 2.7.3).

2.3.3.1 Subordinate structures

Types :

– **Sub-S1** : [When S, S] or [S when S]

Ex: 2.2.1a *Blockout start is enabled when at least one DG is in AUTO-mode and not blocked*

– **Sub-S2** : [if S, S] or [S, if S]

Ex: 2.3.14 *If the operator wants to stop an online, PMS controlled DG, this can be done from the ISC consoles*

– **Sub-S3** : [XS, S] or [S, XS] X∈ {*after, before, while*}

Ex: 2.4.51 *No disconnection is performed before this is satisfied*

– **Sub-S4** : final infinitive

Ex: 2.4.28 *In order to connect SG to the BB...*

– **Sub-S5** : other than Sub-S1 to Sub-S4

a : [if S, S, when S]

b : [S, so that S]

c : [in case S, S]

d : [when S, S, until S]

e : [if S, S, so that S]

2.3.3.2 Nominal expressions (= Ne)

Types :

- **T :** ordinary titles

Ex: 1.1 *Purpose and Scope of Power Management System*

- **T:** : pseudo-titles

Ex: 2.1.1.1 MANUAL :

- **St :** statement, item in an indented enumeration with no subordinate expression

Ex: 2.1.4b *No control at all of DG in question*

- **Sub-St** : statement with a subordinate expression

Ex: 2.1.18 ...change to the next DG in the stand by sequence, if a DG does not start

2.3.3.3 Embeddings (= E)

Types :

- **that S**

  Ex: 2.2.1b *Blocked means that the DG is not available*

- **[which...] S**

  Ex: 2.2.11b *...which on line DG is frequency controlled*

- **[wh...] S relative**

  Ex: 2.2.12b *the DG which is supposed...*

- **infinitive** (not purpose)

  Ex: 2.2.5 *if the former DG fails to start or switch on line*

2.3.3.4 Complex sentences (= cS)

Types :

- **cS+-** : verbal form coordination (including "/") without embedding

  Ex: 2.4.23 *DG's are disconnected and stop*

- **cS-+** : embedding without verbal coordination

  Ex: 2.1.9 *The models require that the DG's are in AUTO mode*

2.3.3.5 Coordination (=C)

The general pattern

$[(C7_i{}^*)\quad x1C_j\qquad x2C_k]$

is assumed, where (C_i) is a recursive and possibly null constituent, and

x1...x2 = *null...{and, or, ","}*

or *either... or*

or *neither... nor*

In Co-S strucures, x2 (with x1 = null) assumes also (in some ad-hoc way) the values of "i.e" and "-".

Types :

- **C1 :** Nominal heads

  Ex: 1.1 *Purpose and Scope of Power Management System*

– **C2** : Predicate nominal

Ex: 1.1.1 ... *is three Diesel Generators (DG) and one Shaft Generator (SG)*

– **C3** : Top level NPs

Ex: 1.1.4 ...*includes synchronization to busbar (BB) and automatic connection of SG to BT/ST*

– **C4** : Left NP modifiers and arguments

Ex: 1.1.3 ... is a *standardized full-automatic start/stop, synchronization, frequency control, loadsharing and black out start* system

– **C5** : right NP modifiers and arguments

Ex: 2.1.3 ... two other modes *AUTOMATIC and SEMI AUTOMATIC*

– **C6** : PPs

Ex: 2.3.1 ... from the MSB or from the PMS

– **C7** : Prep NPs

Ex: 2.1.3 ... to either BB or BT/ST

– **C8** : Verbal phrases (sentence forms included)

Ex: 2.3.24 The PMS will then *automatically up date the plant... and then stop the one in question*

Ex: 2.2.5 ... fails *to start or switch on line*

– **C9** : Other than C1 to C8

Ex: 2.3.3 ... is controlled *either from the MSB or directly on the AE*

2.4 Semantic analysis

Two types of semantic representations are assumed, depending on the type of semantics on which they are grounded: (i) truth conditional semantics and (ii) communicative semantics. Only the representations of the first type are subject to an inferential calculus.

We use here the label *communicative semantics* to refer to different kinds of work in the field of *descriptive* semantics, among which are e.g. historical semantics (Bréal, etc.), semantics of speech acts (Ducrot), and interpretative semantics (Jackendoff, etc.). It is well known that all observations of a communicative semantics have not yet been formalised. This is illustrated, among others, by coordination and plural NPs. If it is possible to classify meanings associated to plural NPs into *distributive* vs. *collective* meanings, we do not know of any published and accurate formal account of these phenomena. In the same vein, coordinations of the type of

a man walks in the park and he whistles

where anaphora and temporal relations are involved, have very recently been treated within dynamics logic, which extends Kamp's DRT (cf. Groenendijk, Stokhof & Beaver, DYANA Deliverable R2.24, 1991), but the state of the art on this point does not allow for an exhaustive account of all the complex problems involved. For this reason, this section concentrates on communicative semantics observations.

Plural NPs

Most NPs are singular. In the 183 T. units of sections 1 and 2, N (NP pl) $\leq$ 65. With one possible exception (2.3.13), all NP[pl] have a distributive meaning; *between* in 2.3.13 introduces a kind of reciprocal meaning.

Anaphora

Anaphora forms and their occurrences are very limited. (The label *anaphora* is reserved here to the non symmetric relation between a referential unit and a pronoun or anaphoric form. In this pattern, definite descriptions, though relevant with respect to the more general problem of coreference, are not considered). Anaphora forms are *that, they, this, these, their, it* and *its*. They corefer with a previous antecedent, and in general coreference is not ambiguous. Appendix D1 resumes observations on anaphora.

Negation

Negative forms and their occurrences are very limited. Negative forms are classified into Neg1 (*no, not*) and Neg2 (*except, without*). The table in Appendix D2 resumes observations on negation, with the exception of coordinate forms. The third column indicates the negated constituent of a Neg1 form.

Verbal forms

With only one exception (namely *could* in 2.3.12), the 228 verbal forms of sections 1 and 2 are in the present tense. But no verbal form expresses *time* related to a particular speech act. Present tense in thus *timeless* (state present or habitual present).

Modals, in V3 forms (total 23), are *can* (6), *must* (3) and *will* (14). *Can* and *must* express their habitual modal meaning. *Will* does not express future time, but a predictive meaning. All V3 *will* forms can be expressed by a V1 form with *will* deletion.

In V4 forms (15), modal forms are *can* (7), *must* (3) and *will* (5). It is here also possible to substitute V1 forms for V4 *will* forms.

Many (7) infinitives (in V5 forms (18)) are used to express *purpose.*

There are only 6 V6 forms. One of them (*does not*) is just a negative timeless present tense. The two progressive constructions (*be ... ing*) and the two perfective ones (*have ... ed*) can be changed to V1 forms. (The last V6 form is the unique *could* exception, cf. above).

Subordinate sentences in nominal expressions (cf. 2.3.3.2) are a subset of subordinate sentences in subordinate structures (cf. 2.3.3.1). The following are attested forms :

x S

with x $\in$ {when, if, after, while, in case, before}

But the variety of syntactic forms hides an invariance in meaning.

if can subsitute for *when* in all cases, with perhaps the exceptions of 2.2.3 and 2.2.4 for stylistic reasons.

when can subsitute for *if* in all cases

when and *if* can substitute for *after, while* and *in case.*

The unique occurrence of *before* is in 2.4.51. The general structure is

[... before V2]$_S$ [neg V2]$_S$

It can be changed to

[... {when, if} V2]$_S$ [V2]$_S$

it appears thus that *when, if, after, while, in case* and *before* subordinate sentences can be reduced to only one semantic type expressing *contingency.*

Semantically, we arrived at the conclusion that in the analyzed text, the semantic relations (R) between subordinate sentences (SubS) and the matrix expression (M) are condensed in the following formulae :

R(M, SubS)

R $\in$ {contingency, result, purpose, time before}

We have thus :

SEMANTIC RELATION	LINGUISTIC FORMS
contingency	when, if, after, while, in case, before+negative
result	so that
purpose	infinitives
time before	until

The syntactic analysis of coordination (cf. 2.3.3.5) reveals two important semantic phenomena : (i) sources of ambiguity, and (ii) choices between paraphrastic structures. (i) concerns alternative analyses as C1 or C3; ambiguity can also be found in C4 and C5. Choices between paraphrastic structures concern mainly C6 and C7, and different levels of verbal phrases coordination (cf.C8). The syntactic sources of coordination combine with well known ambiguous semantic values of *and* and *or*. Coordination is associated with negation in the construction *neither ... nor*, which is present in 2.4.53.

In the analyzed text, the symbol " / " is frequently used as an indicator of conjunction. Its semantic value is very ambiguous, ranging from "and/or" to a distributive meaning related to the *respectively* coordination.

2.5 Lexical analysis

The lexical analysis is centred on verbal entries. Two main points were considered : (i) semantic classification ; (ii) syntactic sub-categorisation and assignment of denotations to the arguments.

Extracted main verbs (i.e. neither modals nor auxiliaries) are presented in Appendix E1. The semantic classification system of Dowty (1979) is resumed and illustrated in 2.5.1. Section 2.5.2 deals with (ii). Appendix E2 presents a list of lexical entries described in terms of the pattern in 2.5.2.

2.5.1 Semantic classification of verbs

Dowty's classification was chosen because it is associated with operative tests relating semantic classes to syntactic-semantic constructions. Another, more compelling reason, is that Dowty's proposal is completely formalised; it uses very few and simple semantic primitives; and, in principle, it could be used in the formalisation of our own proposal. In the following, (i) presents the classification and (ii) the associated tests. In (iii) the classification is partially related to other current terminology.

(i) Verb semantic classification

I	change of state (ch)												-change of state			
II	Definite (Def)								-Definite				(-ch)			
III	Single (S)				-Single (C)				(Ind)							
IV	Ag		-Ag		Ag		-Ag		Ag		-Ag		Ag		-Ag	
V	M	-M	M	-M	M	-M	M	-M	M	-M	M	-M	M	-M	M	-M

<u>Terminological conventions</u>

-definite = indefinite
-single = complex
Ag = Agentive
M = Momentary
-M = Interval = I

(ii) Tests
(0 = no ; 1 = yes)

I *what NP did was V*
1 : change of state
0 : -change of state

II *NP was Ving* entails pragmatically *NP has Ved*
1 : indefinite
0 : definite

III *NP finished V ing*
1 : complex
0 : single

IV *V imp, persuade NP to V*
1 : agentive
0 : - agentive

V *is ∨ing*
1 : momentary
0 : interval

(iii) Terminological near equivalences

- **no change of state = states**
- **indefinite change of state = activities**
- **definite change of state = accomplishment or achievement**

The following table illustrates the result of applying Dowty's classification to selected verbs which express typical domain processes. How to integrate this type of results in the general pattern proposed in the next section remains an open question.

	I	II	III	IV	V
synchronize	ch	Def	C	Ag	I
connect	ch	Def	C	Ag	I
stop	ch	Def	S	Ag	I
start	ch	Def	S	Ag	I
close	ch	Def?	C	Ag	I
select	ch	Def	C	Ag	I
switch	ch	Def	C	Ag	I
command	ch	Ind	?	Ag	I
open	ch	Ind	?	Ag	I
run	ch	Ind	?	-Ag	I

2.5.2 Lexical entries

The main assumptions we make with respect to the lexical data of the PMS document are the following :

(1) The Semantic Representation of a Sentence (SRS) is wholly specified by the syntax and the lexicon (simplified assumption).

(2) Syntactically, technical documents can be characterised as sub-languages with a "simple" syntax.

(3) In technical documents, lexical information (a) strongly determines SRS and (b) depends widely on domain knowledge.

(4) SRSs associated with technical documents present significant variation inasmuch as they are produced by specialists in the field or by people akin to it.

(5) Technical documents are produced for specialists in the field.

With these ideas in mind, the challenging goal we are faced with may be summarized as follows:

- Characterize in general terms the notion of technical lexicon in order to account for SRSs associated with technical documents by specialists, while keeping the distinction between *lexical knowledge* and *encyclopaedic knowledge* (see below).

- Specify a significant subset of lexical entries relevant to the examined documents (see Appendix E2).

Basic elements of a lexical entry

A lexical entry is taken to be constituted by (a) a *lemma* i.e. the dictionary entry; (b) a *syntactic categorisation* (**SC**) of the lemma; (c) a *semantic representation* (**SR**) of the lemma in terms of a semantic predicate together with its arguments; (d) a list of *syntactic frames* (**SFr**) i.e. the list of all the sub-categorization schemas in which the lemma may be involved, where each syntactic category is further specified by its selectional features (**SF**); (e) a list of *equations* relating the semantic representation to each syntactic frame; and (f) a list of *denotational constraints and/or equations* restricting the use of the lemma. Items (a) to (e) define the *generic lexicon*, whereas item (f) defines the *specific domain lexicon* related to the PMS document.

The schematic form of a lexical entry will thus be the following :

• Lemma; SC	(a), (b)
• **predicate** $(A_1, ..., A_n)$ with $n \geq 1$ and $A_i = \langle\theta\text{–role}(x)\rangle$	(c)
• $(C_1, ..., C_n)$ with $n \geq 1$ and $C_i = \langle SC, SF\rangle$	(d)
• $A_i == C_i$ (the denotation of C_i instantiates A_i), ...	(e)
• **x' = y' ; x' –> y', ...**	(f)

A *reading* of the lemma **L; SC** is defined as **<SR, SFr>** (i.e. all equations relating the semantic representation to *one* syntactic frame, cf. (e) above), where for every C in SFr there exists an A in SR such that C == A.

Furthermore, it must be noted that :

- a lemma can incorporate more than one SR and/or SFr
- it is possible to specify more than one reading for each lemma
- an A_i in a specific reading may have no associated C_i
- (C_i) is the notation corresponding to the fact that C_i is an optional constituent

A list of lexical entries is presented in Appendix E2. The information is displayed as follows : occurrence (**Occ**), grammatical category (**GramCat**), arguments of the predicate, additional information or comment (if any). For each occurrence of the lemma, the syntactic data is given on the first line, whereas the semantic information is given on the second line (in bold). The cases where there is semantic information corresponding to no syntactic information are the cases where the context enables a semantic retrieval or reconstruction of the missing argument.

2.6 Monotonicity

One important issue is to investigate if the characteristics of the analyzed text are also to be found in other sections of the same text or in other texts of the same class (See 2.7.2 (ii) and the Representativeness' checker of Fig 1 in 2.7.3).

Section 4 of the same document is being analyzed with this underlying problem in mind. First global results concerning coordination tends to verify some kind of monotonicity:

	N (Tunits)	N(C)
Sections 1 & 2	183	74
Section 4	231	131

Besides the quantitative aspect, the typology of coordination which accounts for coordinate forms in sections 1 & 2, is also valid for section 4.

2.7 Conclusions

2.7.1 Linguistic analysis results

The main feature of the analyzed text is its *simplicity*. The notion of simplicity, although rather intuitive, can be further characterized as follows:

- Within the limits of 'communicative semantics' and with the exception of certain kinds of coordination, the linguistic structures of the text can be accounted for within the framework of most of the existing grammatical formalisms (e.g. categorial grammars, UCG, LFG, HPSG or GPSG), and
- The linguistic structures of the text form a restricted subset of the core structures of the HP-NL Test Suite (Flickinger et al. (1987)).

With respect to the core constructions, some negative characteristics must be noted as well, such as the absence of unbounded dependencies constructions, the absence of some case of cross-categorial coordination, the absence of comparatives and related constructions, etc.

The most salient results are thus the following:

SYNTAX

- Heavy use of coordination (and slash)

SEMANTICS

- Absence of problematic temporal phenomena
- Absence of interrogation
- Very simple negation, which does not involve intricated questions of quantifiers scoping
- All subordinate clauses may fit into a unique semantic representation

LEXICON

- Very few composition (loadsharing, shut-down, slowdown) but lots of quasi frozen expressions: black out, control modes, start system, black out start, start sequence, EMS, MSB, ship handling mode selector, priority sequence, master sequence, standby sequence ...
- Limited cases of derivation: connect/disconnect, load/deload, enable/disable, exit/deexit, start/restart.
- Some cases of category transfer, e.g. between V[Psp] and ADJ, or V[ing] and N or ADJ, as illustrated in Appendix E2.

- The main issue with respect to lexicon is, given some language, the theoretical delimitation between (a) the general lexicon of that language, (b) the general lexicon of a domain and (c) the specific lexicon of a domain. In our case, (a) is the English lexicon, (b) the general lexicon related to power management and (c) the specific domain lexicon related to the PMS described in the PMS document. In Appendix E2, according to § 3.2, the first part of the lexical entries expresses (b) whereas the last one expresses (c). However, the questions remain open, as whether this distinction is the good one (the alternative being that the whole entries do characterize the generic domain lexicon (b)), and as where the domain-specific information should be represented.

2.7.2 Underlying issues

The major general issues underlying text analysis can be resumed by the following points.

(i) Terminology

Though at first glance secondary -even if important- the terminological problem hides a deeper conceptual problem. If, in principle, it is impossible to describe linguistic observations independently of a particular theory, as it is suggested by, for example, Uszkoreit (1987), no theory independent analysis on corpus can be made at all. Here it was taken for granted that this extreme position is methodologically inadequate, and that it is possible to manipulate coherently the labels of a descriptive meta-language in order to describe linguistic material (cf. Bès & Jurie (1989)). But in a complete study, the definitions of the labels used must be carefully presented, ensuring that in this way confusions will not arise in the final result (see for example in § 2 the distinction between 'communicative' semantics and truth conditional semantics, or the status of the dividing line between anaphora and definites descriptions).

(ii) Adequacy of texts

The interest of corpus study is not focused on the analysis of texts as such, but rather on texts as illustrative samples of regular and recurrent phenomena. The issue of the adequacy of texts involves two aspects, quantitative and qualitative.

The quantitative aspect is a classic statistical problem : relation of the lengths of analyzed texts with the number of observed phenomena. (From the *Summary table* of § 2.3.2, the *Summary of occurrences* sections of Appendices C3 to C7 and § 2.6, it is possible to draw information relevant to this point (see the 'representativeness' checker in Fig. 1 § 2.7.3)).

From a qualitative point of view, existing documentation must not always be considered excellent texts to be blindly imitated in all of their features. It is often necessary to consider alternative formulations expressing differently the 'same content'; that is, formulations with different communicative effects. Furthermore, it is well known that any sub-language is not just included in the general language (cf. Kittredge & Lehrberger (1982)) : underlying any sub-language there is always some rule-changing creativity. This is, for example, shown in the analyzed text in the sub-categorization of *synchronize*. To distinguish between acceptable rule-changing creativity and not acceptable deviancy is always a delicate problem. This means that the results of any corpus analysis cannot be used non critically as a direct specification of a planner.

(iii) Input text

The input to the linguistic analysis cannot be raw text. In one way or another, it is necessary to operate with text units one needs to refer to systematically in the course of the analysis. A formatisation of the text in terms of typographic elements is thus necessary. Because this is intended to facilitate subsequent text and sentence analysis, this task is less trivial than it appears at first glance (see Appendix C1).

A much more delicate operation, which is also absolutely necessary to fulfil before the linguistic analysis, is a partial content interpretation of the text, with the help of a specialist and/or in terms of the domain model. The amount and quality of the information required at this pre-analysis step is an open question. That information required for solving some kinds of lexical and syntactical ambiguities will certainly play a crucial role.

(iv) Selection of linguistic constructs

The adequate output of a corpus analysis in the syntactic, semantic and lexical levels, is not a complete parse of each sentence associated to an exhaustive semantic representation. Rather, what is wanted is a characterization of the 'typical' or 'defining' constructs attested in particular classes of texts.

The underlying assumption of a sub-language study is that particular denotational universes introduce constraints in the linguistic forms which can or must be used for expressing them (see Kittredge & Lehrberger (1982), Arrarte et al. (1991)).

The problem that arises then is one of selection of the linguistic constructs that are to be used while proceeding in the analysis. The problem involves at least the following questions: (a) a reusable grammar repository in terms of which the analysis of any text can be performed ; (b) an a-priori selection of constructs in order to capture quantitative macro-constraints in the representations of text units; (c) the discovering of qualitatively 'interesting' phenomena in the analyzed texts in order to specify non trivially the class of texts related to a particular domain.

(v) Manual vs. automatic

How to determine the adequate trade-off between manual and automatic processes is an open question. If existing powerful grammatical formalisms should allow to perform automatically many of the required morphological and syntactic analysis, it is also true that semantic information is needed which lies beyond the scope of existing automatic procedures of analysis.

(vi) Knowledge representation formalisms

KRL formalisms are needed for both domain representation and linguistic knowledge representation.

Linguistic knowledge representations are needed at different and interactive levels (See next section): the Text DB (see Appendix C2), the Text Structures KB (See Appendices C3,C4,C5,C6,C7) and the Lexicon (Specific) KB (See Appendix E2). Again, the main questions remain open: (a) unique formalism for both domain and (the several kinds of) linguistic knowledge, or different, specific formalisms; (b) links between the different kinds of knowledge and/or between different formalisms. The issue of how to discriminate between the leading kinds of KRL formalisms is a crucial one. At least, the following ones must be considered:

- DATR — Gazdar & Evans (1990)
- Conceptual Graphs — Fargues (1989), Fargues et al. (1986)
- TFS — Zajac (1991)
- OBJLOG+ — Chouraqui & Godbert (1989), Chouraqui & Faucher (1990)

2.7.3 General Methodology

Figures 1 to 3 below resume the basic properties of a general methodology of text analysis considered in a global pattern, the target of which is the definition of specifications to the generator.

Fig. 1 presents the general pattern, dubbed TEK (Text Extractor of Knowledge). The Text Indexator component formatizes the input texts (automatically) and assigns relevant content information (See 2.7.2 (iii)). The Text Analyzer component is in charge of the core of the analysis. See in Fig. 2 a more detailed view of this process. The Text Analyzer receives Indexed Texts and parses them automatically in terms of selected linguistic constructs (See 2.7.2 (iv)). It produces in this way a Labelled Text. Sentences at this stage have received a Syntactic Representation associated to a partial Semantic Representation. The latter is completed by the following manual process, the output of which is a Codified Text. In this, the representations of sentences incorporate all and only the syntactic and semantic information that is needed to construct the output knowledge sources.

The last and powerful component of the Text Analyzer is the Repository Knowledge System Manager, i.e. the device allowing to organise and structure relevant information driven form the Codified Text.

The output of the Text Analyzer is three knowledge sources from which the Manager will extract relevant information. This is one of the basic areas (the other being the domain knowledge) where KRL formalisms are crucial (See 2.7.2 (vi)).

Besides the core Text Analyzer component, Fig.1 incorporates other components related to the adequacy issue of texts (See 2.7.2 (ii)), both the quantitative (See the 'Representativeness' checker) and the qualitative one (See the 'Pragmatic' checker).

Fig.3 gives the details of the output of the Manager process of Fig.1. The Manager defines specifications in terms of (a) the three knowledge sources storing information about texts; (b) results obtained from the two checkers; (c) other sources of knowledge concerning the domain, the user, documentation standards, and grammar, lexicon and terminology.

It is assumed that the Generator incorporates two basic components, (a) a text and sentence grammar, and (b) a planner, and that, consequently, the Manager must produce both kinds of specifications.

The Manager will use the Text Structure KB and the Text Specific Lexicon KB in order to produce grammar specifications. On the planner side, the Text DB will furnish the main information underlying quantitative constraints on text units and texts. But the more challenging information that must be given to the planner is the qualitative constraints on the conditions of use of a particular linguistic construct. Main relevant information comes here from the Text Structure KB and the user model. Planner specifications must answer the multiple question of *how to say what to whom,* facing two basic and challenging problems: (i) the choice of the adequate member of a paraphrastic class; (ii) the amount of redundancy and/or of its counterpart, ambiguity, allowed in texts, depending on the communicative goals of the text and the user who is intended to profit from it.

TEK (Text Extractor of Knowledge)

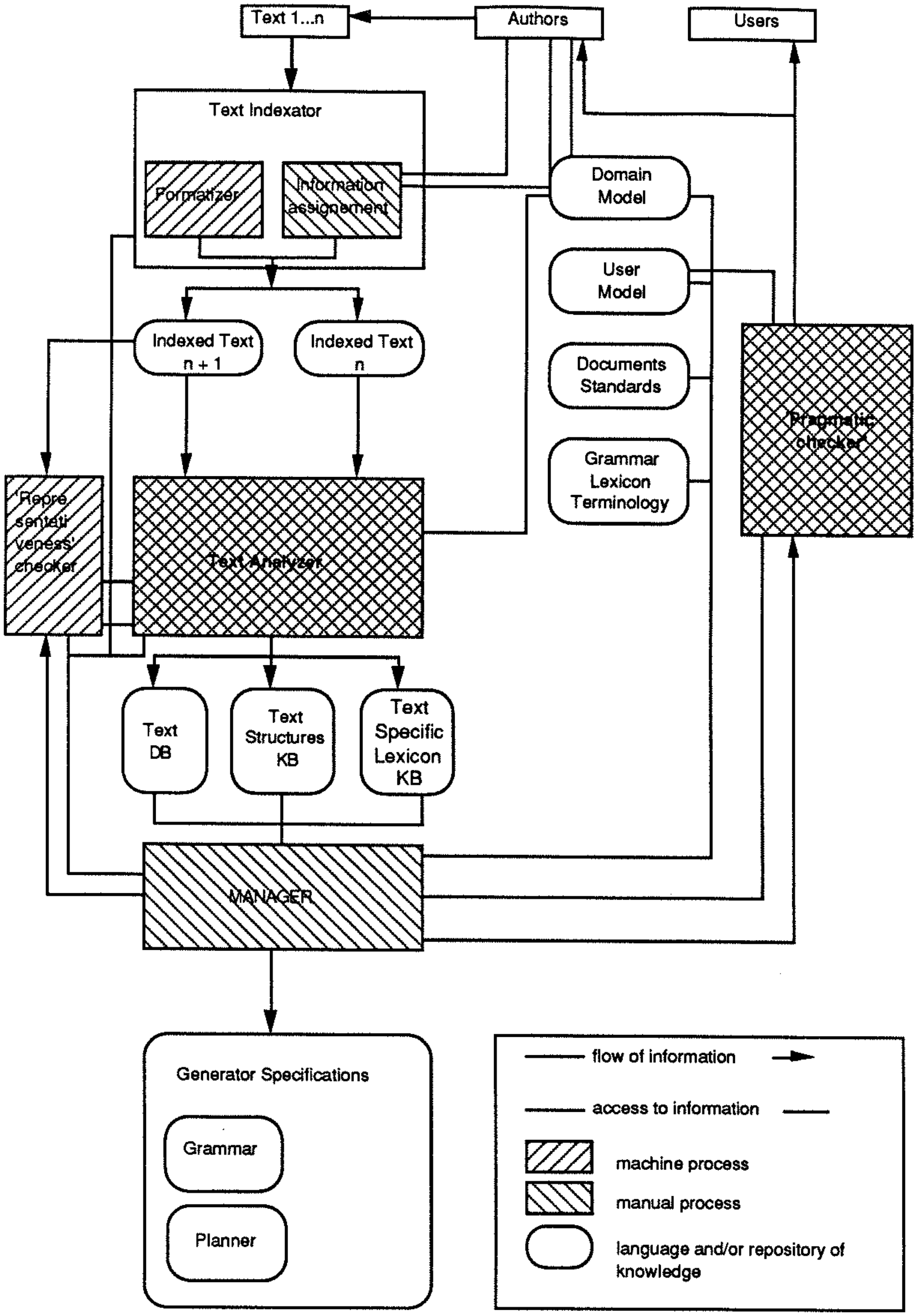


Fig. 1

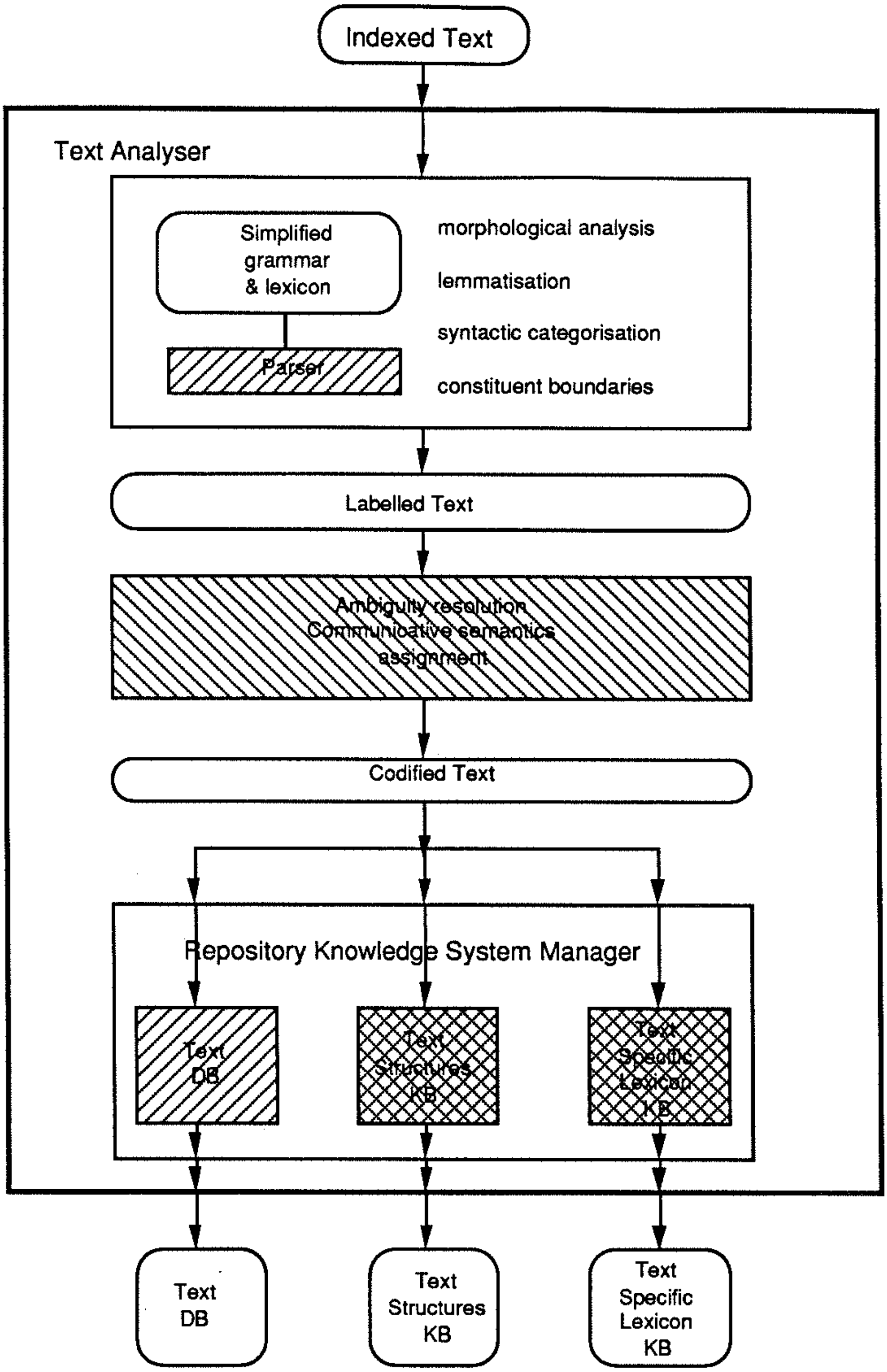

Indexed Text
Text Analyser
Simplified grammar & lexicon
Parser
morphological analysis
lemmatisation
syntactic categorisation
constituent boundaries
Labelled Text
Ambiguity resolution
Communicative semantics assignment
Codified Text
Repository Knowledge System Manager
Text DB
Text Structures KB
Text Specific Lexicon KB
Text DB
Text Structures KB
Text Specific Lexicon KB

Fig. 2

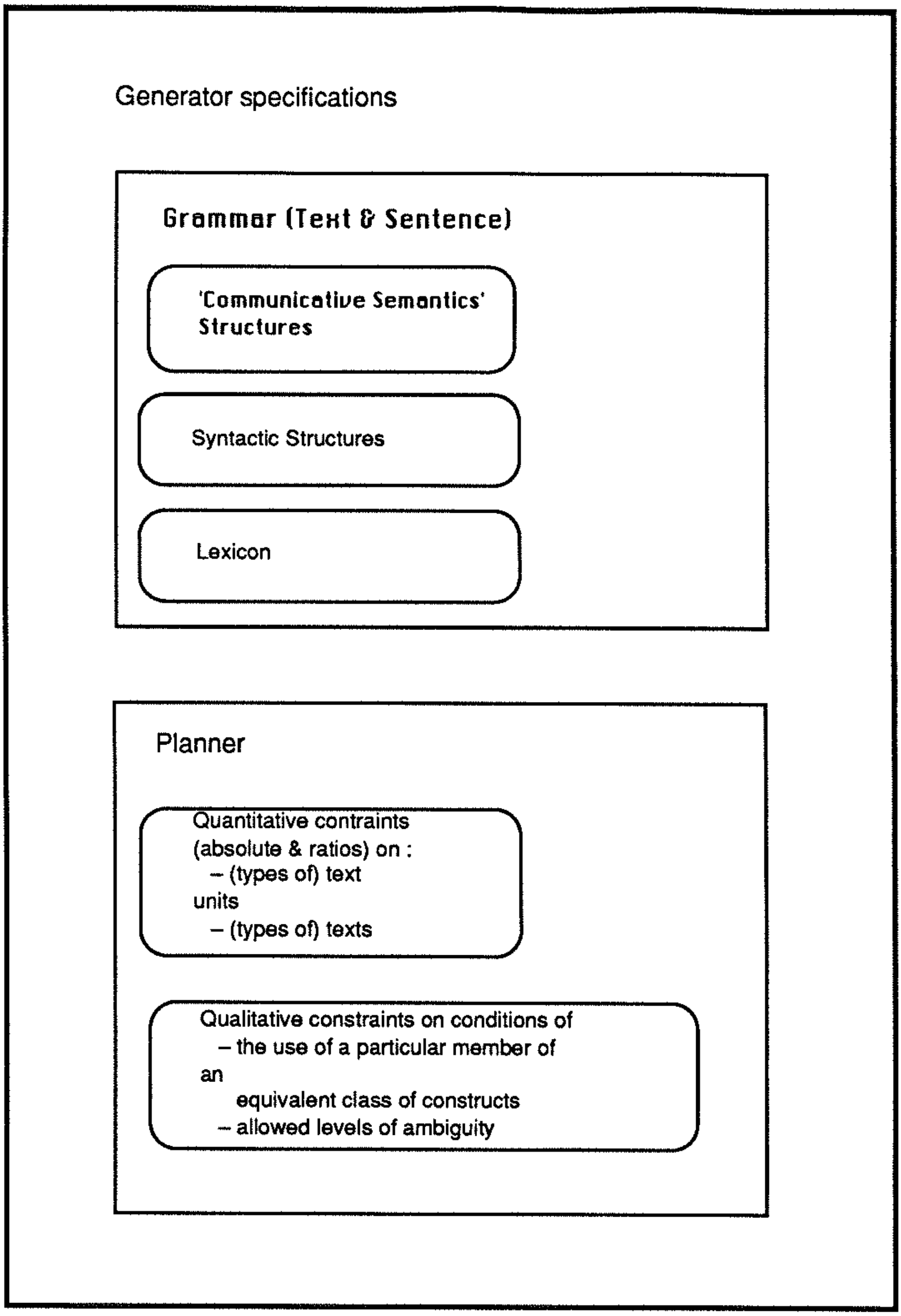

Generator specifications
Grammar (Text & Sentence)
'Communicative Semantics' Structures
Syntactic Structures
Lexicon
Planner
Quantitative contraints
(absolute & ratios) on :
– (types of) text
units
– (types of) texts
Qualitative constraints on conditions of
– the use of a particular member of
an
equivalent class of constructs
– allowed levels of ambiguity

Fig. 3

2.7.4 - References

Arrarte, G., M. Muñoz, T. Redondo, M. Sobejano & I. Zapata (1991) *A Management Tool for Test Corpora*, MT Evaluation Workshop, Vaud (Switzerland), April 1991.

Bès, G.G. & P.F. Jurie (1989) *UCG grammars; the control of their descriptive adequacy*, in ESPRIT Project 393 ACORD Technical Documentation, 1989, pp.207-214.

Chouraqui, E. & E. Godbert (1989) *Représentation des descriptions définies dans un réseau sémantique*, Marseille, GRTC, 1989.

Chouraqui, E. & C. Faucher (1990) *OBJLOG+, un langage de schémas étendu et extensible*, Marseille, GRTC, 1990.

Dowty, D.R. (1979) *Word Meaning and Montague Grammar*, Dordrecht, Reidel, 1979.

Fargues, J. (1989) *CG information retrieval using linear resolution, generalization and graph splitting*, IBM Patris Scienfific Centre, 1989.

Fargues, J., M.C. Landau, A. Dugourd & L. Catach (1986) *Conceptual Graphs for semantics and knowledge processing*, IBM J. Res. Dev., 30, 1, p.70-79.

Flickinger, D, J. Nerbonne, I. Sag & T. Wasow (1987) *Toward evaluation of NLP systems*, Palo Alto, Ca., Hewlett-Packard Laboratories, 1987.

Gazdar, G & R. Evans (eds.) (1990) *The DATR Papers*, School of Cognitive and Computing Sciences, Brighton, Sussex, 1990.

Kamp H. (1981) "A theory of truth and semantic representation", in Groenendijk, Janssen & Stockof (eds.) *Formal Methods in the Study of Language*, Mathematical Centre Tracts, Vol. 1, Amsterdam, 1981, pp.277-322.

Karlsson, F. (1990) *Constraint Grammar as a Framework for Parsing Running Text,* Coling 90.

Kittredge, R. & J. Lehrberger (eds.) (1982) *Sub-language; Studies of Language in Restricted Semantic Domains*, Berlin, de Gruyter, 1982.

Uszkoreit, H. (1987) *Word Order and Constituent Structure in German*, CSLI Lecture Notes, 1987, n°8.

Zajac R. (1991) *Notes on the Typed Feature System*, project Polygloss, IMS-CL/Ifl-AIS, 1991 (Draft).

3 EVALUATION OF MODELS

3.1 Presentation of models

3.1.1 Unification Categorial Grammar

3.1.1.1 A brief introduction to Unification Categorial Grammar (UCG)

In UCG, the basic linguistic unit is a *sign* which includes phonological, syntactic, semantic and ordering information. A sign may be represented either by a complex feature structure e.g.

$$\left| \begin{array}{l} \text{pho: STRING} \\ \text{synt: CAT} \\ \text{sem: SEM} \\ \text{order: ORDER} \end{array} \right|$$

or by a sequence of feature values separated by colons e.g. STRING: CAT: SEM: ORDER. The phonological field of a sign contains its orthographic string. The syntactic field is categorial i.e. it can be either basic (with value **s**, **n** or **np**) or complex. A complex syntactic field is of the form **C/Sign** where **C** is a syntactic field and **Sign** is a sign called the *active* sign; **Sign** is also referred to as the *active part* of the including sign. Moreover, any basic category can be assigned some morphosyntactic information. For instance, **s[fin]** denotes the category s(entence) with morphology feature value *finite*. The semantic field contains the semantics of the expression. The semantic representation language, called *InL* (for Indexed Language, cf. Zeevat (1986)), is a linear version of Discourse Representation Theory (Kamp (1981)). As in most unification-based grammars, the semantics of any expression results from the unification of the semantics of its sub-parts. Thus the semantics of an expression is constructed compositionally via unification, and the semantic representation of any sentence in UCG is simply a further instantiation of the semantics associated lexically with one element of the sentence. Finally, the order field is a binary feature with value either **pre** or **post** which constraints the applicability of grammar rules.

Grammar rules in UCG are of two types: binary and unary. Binary rules include forward and backward functional application. The rule of forward application is stated below (the rule of backward application is similar except that the argument sign precedes the functor sign and the order values of the argument and of the active sign are **post** rather than **pre**).

$$\left| \begin{array}{l} \text{pho: Wf} \\ \text{synt: Cf/} \left| \begin{array}{l} \text{pho: Wa} \\ \text{synt: Ca} \\ \text{sem: Sa} \\ \text{order: pre} \end{array} \right| \\ \text{sem: Sf} \\ \text{order: Of} \end{array} \right| \quad \left| \begin{array}{l} \text{pho: Wa} \\ \text{synt: Ca} \\ \text{sem: Sa} \\ \text{order: pre} \end{array} \right| \quad \text{--->} \quad \left| \begin{array}{l} \text{phon: Wf + Wa} \\ \text{synt: Cf} \\ \text{sem: Sf} \\ \text{order: Of} \end{array} \right|$$

Unary rules are of the form a --> b where a and b are signs. Unary rules are used for the treatment of unbounded dependencies, syntactic forms of type-raising and subcategorisation for optional modifiers.

The following is a sample derivation of the parsing of sentence *The generator stops*:

the

W:
(C/(W: C/the+W1: np[nom or obj]: b: O): [a] S: O)
/(W1: noun: [b] R: pre):
[a] [[b] R, S]

As quantified NPs are treated as typed-raised terms, the determiner introduces type-raising (the polymorphic nature of UCG categories allowing to have a single representation for NPs, regardless of their syntactic context). *The* combines first with a noun which has phonology **W1** and semantic index **b**. The semantics that results from such a combination is a conjunction, the first conjunct of which is the semantics of the noun. The second conjunct is the semantics of the resulting NP's argument (i.e. the verb).

generator

generator:
noun:
[x] generator(x)

The combines with *generator* via forward application to give *the generator:*

W:
C/(W: C/the+generator: np[nom or obj]: b: O): [a] S: O:
[a] [[b] generator(b), S]

The generator combines with *stops* via forward application again to give *the generator stops:*

stops

W+stops:
sent[fin]/(W: np[nom]: X: pre):
[e] stop(e,X)

the+generator+stops:
sent[fin]:
[e] [[b] generator(b), stop(e,b)]

This (simplified) final semantics must be read as "there is a stopping event **e**, of which **b** is the patient, and **b** is a generator".

Let us present now a sample derivation of the generation of sentence *The generator produces power*. The input to the generator is the following (for more details, cf. § 2):

```
head( produce(g,p),
      arg([specifier(the, head(generator(g), [], [])),
            head(power(p), [], [])]),
      adj([]) )
```

Roughly, generation will proceed as follows. Suppose the goal sign **Sign0** has category **sent[fin]**. First, the semantics coresponding to the head of the clause (i.e. **produce(g,p)**) is extracted and a sign **Sign1** is created with semantics **produce(g,p)**, which becomes the following after lexical access:

W1+produces+W2:
sent[fin]/(W1: np[nom]: g: pre)/(W2: np[obj]: p: post):
[e] produce(g,p)

Sign1 must then be reduced to **Sign0** with category **sent[fin]**. At this stage, the remaining input semantics is:

arg([specifier(the, head(generator(g), [], [])),
head(power(p), [], [])])

To generate the arguments of **Sign1**, we may then generate on the basis of **head(power(p), [], [])**, the whole process recursively going on as above:

power

W:
C/(W: C/the+W1: np[nom or obj]: b: O): [a] S: O
[a] [power(p), S]

produces power

W1+produces+power:
sent[fin]/(W1: np[nom]: g: pre):
[e] [power(p),produce(g,p)]

Generation goes then on on the basis of **specifier(the, head(generator(g), [], []))**:

generator

generator:
noun:
[g] generator(g)

the generator

W:
C/(W: C/the+generator: np[nom or obj]: g: O): [a] S: O:
[a] [[g] generator(b), S]

the generator produces power

the+generator+produces+power
sent[fin]
[generator(g),power(p),produce(g,p)]

At this stage, there is no more input semantics, and the successive reductions performed between the signs has ended up in the goal category **sent[fin]**. Thus, the sign corresponding to the string *the generator produces power* can be taken as the goal sign **Sign0**.

As there have been many related publications, we will not put emphasis on the presentation of the model. For more details, the reader is referred for instance to Zeevat, Klein & Calder (1987), Calder, Klein & Zeevat (1988) and Moens, Calder, Klein, Reape & Zeevat (1989). However, we would like to insist upon two points:

- Formally, the objects manipulated by the grammar have well-known properties, in as much the basic data structure of the grammar are *directed acyclic graphs*. Therefore, UCG actually works in a PROLOG implementation of the PATR-II formalism (cf. Shieber et al. (1983)), which was explicitly designed as a language within which the declarative portion of any unification grammar can be stated and computationally implemented.

- The *lexicon* is the central component of the grammar (a feature shared with Meaning Text Theory, but in a more radical way (cf. Karttunen (1987)). Within UCG, the structuring of the lexicon is achieved by means of PATR-II templates and lexical rules. Moreover, high-level generalizations over the lexicon are introduced via data-typing and sorted logics (cf. Calder & Lindert (1987), Calder, Klein, Moens, Reape (1988) and Moens, Calder, Klein, Reape, Zeevat (1989)).

3.1.1.2 - Generation in UCG

Any generation system is usually supposed to comprise at least the following parts:

- Underlying program
- Text planner
- Linguistic component

We shall examine the main features of UCG with respect to these three parts, as they were actually implemented in the ACORD system (*The Construction and Interrogation of Knowledge Bases Using Natural Language Text and Graphics*).

3.1.1.2.1 Underlying Program

The purpose for which language is generated naturally influences the type of generation required. The ACORD generator was required to provide answers as part of an interactive system: it was answering questions about an underlying knowledge base through short answers (single sentences). It was used only to phrase the content of a short-answer response to a question whereas other (nonlinguistic) components of the system were determining content.

Generation in ACORD had to be performed in the three ACORD languages (English, French and German) using the same grammar formalisms and grammars employed in parsing: UCG for English and French, and LFG for German. The grammatical coverage of the generators included all the natural language phenomena handled by the corresponding parsers.

Whereas the generation of a sentence expressing the content of some semantic expressions depends necessarily on a language-specific (and formalism-dependent) module, KB querying and the response semantics construction are tasks which are not language-specific and which are thus specified in common modules which are shared by the three individual language generators.

With respect to content determination and planning, it must be noticed that the semantics of the answer to be generated is not generated from scratch, inasmuch the system combines the semantics of the question posed by the user and the KB answer according to the specification of InL.[1] At this

[1] In fact, it is not InL, but SYNInL, as will be explained below.

stage, several decisions are made on defining "what to say" and "how to say it". (For instance, one type of planning ACORD focuses on is NP planning i.e. determining the form a particular NP is to be realized as, which depends on the dialogue history as well as the KB). But these decisions are heavily guided by the form of the question itself.

On the contrary, HYPERDOCSY belongs to an non-interactive application, where the generator must be able to determine content as well as phrasing (and phrasing not only of paragraph-length text, but of (whole parts of) technical documents). The task is therefore much more complicated than in the case of ACORD.

3.1.1.2.2 Text Planning

1 Use of DRT (InL) as semantic representation language

All natural language components in the ACORD system use InL as semantic representation language, which is based upon Kamp's DRT with the following important additions. First, every expression or formula in InL has a particular variable associated with it, called its *index*. The index has two main functions. First, it serves as a marker of the object or event to which the rest of the expression refers. Second, every variable is *sorted* i.e. it contains information about the properties of the object it describes, such as whether it is an event, a process, a physical object, a human, etc.

InL has two logical connectives **and** and **imp**, representing conjunction and implication, respectively. Another construction named **set** allows for the representation of non-singular objects.

A further distinction is introduced when comparing the semantic representations produced by the parsers with those required by the rest of the system. As natural language includes devices for referring to the entities mentioned in a discourse such as anaphoric pronouns and definite descriptions, the parsers produce information that allows a central component, the *Resolver*, to determine the possibilities of coreference. This additional information is incorporated into an InL expression in the form of *occurrence information* or *lists*, stating for every element which may be coreferential with some other element properties relevant for determining coreference.

InL expressions which incorporate such information are referred to as *unresolved* InL and InL expressions where this information has already been used to determine coreference (and thereafter removed) are referred to as *resolved* InL.

The grammatical systems of ACORD allow the relation of InL expressions with individual sentences in French, English and German by the combination of partially specified semantic representations using PROLOG's term unification. Such expressions are referred to as *canonical* with respect to the grammar in question. As with any other logical language, there will be many InL expressions which are logically equivalent. For instance, a conjunction has the same meaning regardless of the order in which the conjuncts are stated. However, the InL expression associated with a sentence will have a particular form determined by how it was constructed, and this form may differ *syntactically* from the other logically equivalent expressions. A consequence of this is that there are InL expressions which are logically equivalent to other expressions associated with strings of a language, but which differ in their form. Such expressions are referred to as *noncanonical*. Simplifying a lot, let us assume for instance that in some grammar G the verb *produces* is lexically subcategorized as:

sent[fin]/(W1: np[nom]: X: pre)/(W2: np[obj]: Y: post)

In that case, the following (simplified) representation of the sentence *The generator produces power* would be canonical with respect to the grammar G:

[e] [[the, generator(x)], [power(y), produce(e, x, y)]]

because it would correspond to the following analysis:

[[the generator] [produces power]]

On the contrary, the representation

[e] [power(y), [[the, generator(x)], produce(e, x, y)]]

would correspond to the analysis

[[[the generator] produces] power]

and would thus be noncanonical i.e. not derivable under grammar G. In other words, there is a direct relationship between the syntactic shape of a semantic formula and the derivational history of the corresponding string.

For the structures the Dialogue Manager (DM) produces it cannot be guaranteed that they are canonical with respect to a given grammar. The existence of noncanonical expressions creates serious difficulties for generation, as the problem of determining whether two syntactically distinct InL expressions are logically equivalent under laws such as commutativity and associativity is factorial in complexity. This has lead in ACORD to the integration of a planning component into the Dialogue Manager and to the definition of an intermediate structure (SYNInL instead of InL) used for generation only, which abstracts away from the derivational information reflected in the linear ordering of the input formula (cf. §2.2.2 below).

Enriched with syntactic information, DRT seems thus to be a possible candidate as a semantic representation language.

Moreover, with respect to pronominalisation, we find the following argument in favour of DRT in Hovy (1990):

> One promising approach to handling pronominalisation is to use Discourse Representation Structures (DRSs) from the Discourse Representation Theory of Kamp (1981) in RST [Rhetorical Structure Theory] paragraph trees. A preliminary description of such use of DRSs is reported in Hovy (1989). Relevant information about each entity mentioned in a clause can be captured in a DRS in the normal way, and the structure can then be propagated upward in the RST tree during tree traversal (just before sentence generation), from nucleus to relation to satellite, where it determines pronominalisation in the satellite clause and merges with relevant information from the satellite. Further propagation proceeds recursively.

Open questions remain : how does pronominalisation relate to the paragraph structure tree ? if DRSs are incorporated into an RST tree, do they provide acceptable pronominalisation ? what are the rules for DRS propagation in the tree ? But, if possible, an integration of DRT and RST (e.g. Mann (1984)) would be interesting.

2 General Architecture

The generation architecture in ACORD consists of three common modules and three language generators, one for each ACORD language. The common modules are located in the Dialogue Manager. They produce the semantic expression of the answer to be generated and make decisions on "what to say" and "how to say it" depending on the state of the KB and the discourse context. The three common modules are :

- The InL -> SYNInL module
- The merger
- The planner

For the sake of generation (i.e. multilingual generation comprising noncanonical input) a representation is required which allows the encoding of syntactic information in addition to semantic information (see Gardent et Plainfossé (1990) for further details). Standard InL, being purely a semantic representation language, is inadequate for encoding this syntactic information. Instead, SYNInL consists roughly of four types : *heads*, *complements*, *modifiers* and *specifiers*. These categories are ideal for attaining a level of language-independence in linguistic description and are general enough that it is reasonable to expect that such X-bar representations can be mapped onto language-dependent surface syntactic structures. However, the language generators are free to realize both scope and surface syntactic structure in any way which is consistent with the SYNInL specification.

As a consequence of the use of SYNInL as a kind of deep structure, noncanonical input is no longer a problem, because the generation algorithm does no more rely on canonical input but on well-formed SYNInL. Omitting the syntactic information usually contained in SYNInL, our previous example (§2.2.1) will be represented as follows:

```
head( produce(g,p),
      arg([specifier(the, head(generator(g), [], [])),
           head(power(p), [], [])]),
      adj([]) )
```

The use of SYNInL has two other consequences. First, since deep structures contain syntactic information, they are a good candidate for the necessary interface between planner and generator. This syntactic information concerns among others the voice (active/passive), the expression of condition (*when*, *if*, *since*, *while*, *because*, ...), the noun quantification and determination (singular/plural, definite/indefinite, demonstrative, cardinality, negation, implication, ...), the type of pronouns (personal, relative, reflexive, possessive), the conjunction/disjunction of objects, events and adjuncts. A second advantage of deep structures is that because they are language independent, they allow for language-independent generation.

The answering process consists of the following steps:

- The question is parsed. The output is the InL representation of the question with occurrence information. This is called the *InL-Q-Occ*.
- The *InL-Q-Occ* is transformed into a *SYNInL-Q-Occ* by the InL -> SYNInL module. The *SYNInL-Q-Occ* is the semantic representation of the question in SYNInL.

- The resolver resolves the *InL-Q-Occ* into the *InL-Q-Resolved*. This step is necessary since all anaphoric expressions must be resolved before querying the KB.

- The following steps are performed in parallel:

  - The *InL-Q-Resolved* is passed on to the KB-TP (Theorem Prover) complex which provides a *KB-Answer*. The KB-Answer is *not* an InL expression.

  - To get a representation of the user's question in terms of SYNInL:
    1. the InL -> SYNInL module maps the *InL-Q-Resolved* into the *SYNInL-Q-Resolved*
    2. the *SYNInL-Q-Occ* and *SYNInL-Q-Resolved* are merged into a *SYNInL-Q*.

- The merger module takes as input the *SYNInL-Q* and the *KB-Answer*. Depending on the type of questions asked, the merger makes decisions such as: what kind of affix is needed, what type of NP-planning is necessary, what kind of answer is expected, and what type of processing can be done on this answer. It calls the planner in order to process all the NPs appearing in the question, as well as the KB-Answer which is transformed into an appropriate SYNInL expression (generally an NP). The output of the merger is a well-formed SYNInL expression: *the SYNInL-Answer*.

- The planner, which is called by the merger, takes the whole SYNInL representation of the question, the name of the current language, and an indication of how the SYNInL has to be modified as input. Using the latter information, the planner decides whether it has to modify the verb phrase, the specifier of an NP or a whole NP. For pronominalisation and the distinction between definite and indefinite descriptions it makes use of the *resolver db*. To produce complex NPs the planner communicates with the KB. For deictic expressions like *this truck*, the planner uses the information about visible objects by asking the DM (which in turn asks the graphic component).

- The *SYNInL-Answer* is the input to the language generator of the current language. The selected language generator generates the final answer. The *SYNInL-Answer* is also used to update the *resolver db* to allow the user to reference by pronouns to objects mentioned in the answer.

The whole architecture is presented in the Figure below.

3 Scope of UCG planning

Three different types of queries are handled in ACORD :

- Yes/no questions
- Wh-questions (e.g. *who, what, where*, etc.)
- Hm-questions (i.e. *how much* and *how many*)

Briefly, the information given to the planner consists of the name of the language and a description of the type of answer together with the answer itself.

In the ACORD framework, ellipsis is not handled, since exactly one sentence is produced as an answer. As said before, the content of this sentence consists of a part of the semantic content of the question and the answer provided by the KB. Full NPs in the input question may be replaced by pronouns and both pronouns and demonstratives replaced by resolved names.

The planner itself can be seen as consisting of three sub-planners, one for verb phrases, one for NPs and one for modifications. When calling the planner, the merger first selects the appropriate sub-planner on the basis of the given SYNInL expression. In the ACORD system this SYNInL expression always corresponds to a sentence, but in a more complete system the planner could also be called recursively to plan several sentences.

The planner does not make decisions about the best verb to choose. However, some decisions about passivization, negation and general information about the arguments of a verb are regularly handled by the planner. Within the ACORD lexicon verbal predicates may only take arguments which refer to objects. This means that there is no planning for arguments which denote events or states, i.e. verbal or sentential complements. Consequently only two types of predicates are distinguished: the copula, which only takes a subject and a noun phrase or prepositional phrase as complement, and all other verbs.

The NP planning component is responsible for providing the best expression for NPs. It uses the dialogue history as well as KB knowledge to decide whether to adopt a pronominalization strategy, or to find a non-pronominal description for the NP under analysis. The NP planner must be provided with enough information to decide whether and which kind of pronominalization is allowed, and whether a name could be used instead of a pronoun where such an option is available. It decides also when to use demonstratives, definite or indefinite articles, and whether a complex description should include relative clauses and modifications. In addition the planner decides which objects should be highlighted on the screen.

The output of the NP planner is a fully specified SYNInL expression, a possible extension of the list of objects to highlight on the screen, a possible extension of the list of local antecedents, an initialization of the subject gender in case the NP corresponds to the subject, and a possible change of the information corresponding to the answer in the event that the NP planner has produced the NP for the answer.

The modification planner can be called either in the context of a verb phrase or in the context of an NP. In the latter case it is assigned the discourse referent of the NP as an argument. The modification planner works on all different types of modifications: verb phrase negation, prepositional phrases, relative clauses, adjectives and adverbs. With respect to pronominalization, it is clear that the options available depend on the adjunct itself. Within the current system personal pronouns may not be generated in the scope of a preposition, and adjunct pronominalisation is not allowed.

To summarize, the planning module is obviously not complete. Nevertheless its design is general enough to allow the incorporation of additional rules and to adapt it to other representations similar to SYNInL. It indicates what the general sources of knowledge are that such a planner would need:

- The element in focus (in a query or in general)
- Accessible antecedents for pronominalisation
- Possible definite and indefinite descriptions
- Objects which can be referred to with demonstratives

The planner also demonstrates how planning can be done for several languages with a minimum of language-specific information (for details, see Kohl et al. 1989)).

Language-specific dependencies concerning gender and the function of NPs could be reduced still further by adopting a slightly different architecture concerning the update of the dialogue history. In this case, the generators would first generate a semantic representation which would then be resolved for dialogue history purposes. Currently, the planner can directly update the dialogue history because it completely decides what type of NP to generate and therefore, indirectly, makes decisions about surface syntacic structure. This will in principle cause difficulties with reflexivization strategies (which are dependent on syntactic factors such as c-command domains) and lexicalisation strategies for verbs. It might also be the case that the planner produces a SYNInL formula which is not resolvable according to the binding theory of a particular grammar (this seems to be more a problem for English than for both French or German).

With respect to HYPERDOCSY, attention remains to be paid to sentence content organisation within text structure, and in particular with respect to (sentence) coordination. The planner should be able to deal with strong vs. weak coordination, i.e. with the fact that most markers are so ambiguous as to be almost meaningless. For instance, *and* can be used to link the elements of most, if not all, rhetorical relations. It is a strong marker of only a few of these relations and a extremely weak marker of the rest, where it tends to mark not a rhetorical relation between the elements that it is linking, but merely the fact that they are part of the same piece of discourse (Gleitman (1965), Lakoff (1971)). Thus it is necessary to determine when the parts of a relation should be realised as a single sentence, and when as separate sentences; and if the parts are realised in a single sentence, when two clauses should be related hypotactically (via subordination or embedding) or paratactically (via coordination). There is a of course a counterpart of this problem within the NP, which the UCG planner should be extended to account for.

3.1.1.2.3 Linguistic Realisation

1 Unification grammar

In the field of *functional* grammars there are the so-called *unification grammar* formalisms (as opposed e.g. to Functional Systemic Grammar (Halliday (1985)[1])). Such unification grammar formalisms (see Shieber (1986) for a good introduction) include McKeown et al.'s FUF formalism (which is close to Kay's Functional Unification Grammar (Kay (1984)), the logic-based formalism of van Noord (which is close to Pereira & Warren's Definite Clause Grammars (Pereira & Warren (1980)) and the Segment Grammar of De Smedt (which is derived from earlier work of Kempen (1987)), all of which are described in Dale, Mellish & Zock (eds.) (1990). In the latter book, we find the following argument in favour of the use of unification grammars for generation:

> The fact that unification is commutative and associative means that the information computed about a phrase (for instance, as a result of taking into account successive aspects of its semantic structure) does not depend on the order in which that information arrives. This makes unification grammars attractive computationally: order independence means a flexibility of operation and leads to a system that satisfies some of the prerequesites of bidirectionality.

According to Appelt (1983):

> Unification grammars are particularily well-suited for language generation because they allow the encoding of discourse features in the grammar. A functional description can be constructed incorporating these features, and the syntactic details of the final utterance can then be specified through unification with the grammar FD [Functional Description]. The process that constructs the text FD can treat it as a high-level blueprint fleshed out by unification thereby relieving the high-level process of the need to consider low-level grammatical details.

1 Although Unification Grammar and Systemic Grammar share many ideas.

But Appelt mentions also the inefficiency of the unification algorithm as a serious problem (it is a non-deterministic process), thus trying to minimize the number of alternatives that ever have to be considered by the system, by means of an interaction between the grammar and the deep generation component. Ritchie (1986) shows that the computational properties of FUGs make general operations NP-complete. Despite these points, McKeown & Paris (1987) are able to achieve processing times similar to that of McDonald's (1980) MUMBLE (TAG formalism) in a reimplementation of FUG.

Belonging to the family of unification grammars, UCG seems to benefit from the same advantages and disadvantages. The general backtracking regime characterising the generation algorithm means that failure at a first attempt to generation might induce the recomputation of partial results. Perhaps the use of a chart could contribute to enhance generation efficiency (cf. Shieber (1988), Calder, Reape & Zeevat (1989) and Gardent & Plainfossé (1990)).

2. ACORD's contribution to the state of the art

Before ACORD, little attention has been paid to the basic *algorithmic* problem of generating a string from a semantic representation according to the syntactic, semantic and morpho-syntactic constraints encoded in the grammar. Only recently, computational linguists have begun to investigate *abstract generation algorithms* i.e. to develop generation algorithms for well-defined classes of grammars which can be shown to be *correct* and *complete* with respect to that class of grammars (Shieber (1988), Shieber et al. (1989)). By *correct* is meant that a generator will not assign a string to a semantic representation which is not logically equivalent to a semantic representation assigned to the string by the grammar. By *complete* is meant that the generator will assign a string to every semantic representation which is logically equivalent to the semantic representation of some string in the language. [1]

In Calder, Reape & Zeevat (1989) and in Gardent & Plainfossé (1990) is to be found a complete description of the generation algorithm in UCG. Apart from the fact that it can deal with non-canonical input, there are two points worth noting about it: first, it permits head-driven generation and second, it provides syntactically guided lexical access. Both these facts enhance efficiency. Head-driven generation is more efficient than functor-driven generation (symplifying somewhat, any semantic functor is also a syntactic functor in UCG) because it starts by generating the most syntactically constraining element, the head. By contrast, in functor-driven generation, identification of a semantic functor often turns out to return a very general functor at the syntactic level: a determiner for instance. Syntactically guided lexical access is clearly more efficient than any purely semantic one. This is particularly true in all cases (which are plentiful) where the semantics is poorly instantiated as, for instance, when searching for a determiner, a clitic or some anaphoric expression.

---

[1] These definitions are taken from Kohl et al. (1989). In Gardent & Plainfossé (1990) is to be found a somewhat different version:

> A generator is said to be *correct* if given two semantic representations R1 and R2 which are not semantically equivalent, R1 and R2 do not generate the same string. A generator is said to be *complete* if any two semantically equivalent representations generate the same set of strings.

These definitions are obviously stronger than the ones above. In particular, Gardent & Plainfossé (1990)'s requirement on correctness is too strong with respect to ambiguity: according to them, a generator would not be able to produce ambiguous sentences, whereas it would be according to Kohl et al. (1989)'s definition.

Going on with our previous example, functor-driven generation would lead to the following steps in lexical access (where the syntactic functors are accessed to first):

1. power
2. the
3. generator
4. produces

whereas head-driven generation would lead to the following steps (where SYNInL heads and specifiers are accessed to before arguments and adjuncts):

1. produces
2. power
3. generator
4. the

The input to the generator is thus a SYNInL structure, where the central notion is that of a syntactic head. SYNInL elements are structured to reflect the thematic dependencies between head, complements and adjuncts and the generation algorithm first uses the semantics of the head to generate a syntactic functor on the basis of which arguments are then non-deterministically generated. With respect to this process, three problem appear, which the solutions to are discussed in detail in Calder, Reape & Zeevat (1989) and Gardent & Plainfossé (1990) and will not be reproduced here:

- type-raised NPs and PPs

Not all type-raised NPs (and PPs) are problematic. Non lexical NPs such as *the cat* or *every man* are functors semantically as well as syntactically and thus conform to the assumption on which the algorithm is based. Problems arise however with proper nouns such as *Harry* or *Johan* because in UCG these NPs are syntactic functors but semantic arguments.

- identity semantic functors

As the name suggests, identity semantic functors are syntactic functors which do not contribute any semantic information - their semantics can be thought of as the identity function. Examples from English are complementizers and case-marking prepositions. From the point of view of generation identity semantic functors are clearly problematic: there is no trace in the input of the phonology that has to be generated.

- adjuncts

In standard UCG adjuncts are not subcategorized for - it must therefore somehow be decided when to generate them.

3.1.1.3 References

Appelt (1983) TELEGRAM: A Grammar Formalism for Language Planning, *Proceedings of ACL.*

Baschung (1990) *Grammaires d'unification à traits et contrôle des infinitives*, Nouveau Doctorat, Université Blaise Pascal Clermont II.

Bès & Gardent (1989) French Order without Order, *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, UMIST, Manchester, 10-12 April 1989, pp.249-255.

Bourbeau, Carcagno, Goldberg, Kittredge & Polguère (1990) Bilingual Generation of Weather Forecasts in an Operations Environment, in *Proceedings of the 13th International Conference on Computational Linguistics*, Vol.2, Univ. of Helsinki, Helsinki, 20-25 August 1990, pp.90-92.

Calder, Klein, Moens & Reape (1988) *General Constraints in Unification Grammars*, ACORD Deliverable T1.6, Centre for Cognitive Science, University of Edinburgh.

Calder, Klein & Zeevat (1988) Unifiation Categorial Grammar: A Concise, Extendable Grammar for Natural Language Processing, in *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, 22-27 August 1988, pp.83-86.

Calder, Reape & Zeevat (1989) An Algorithm for Generation in Unification Categorial Grammar, in *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, UMIST, Manchester, 10-12 April 1989, pp.233-240.

Calder & Lindert (1987) The Protolexicon: Towards a High-level Language for Lexical Description, in Klein & Van Benthem (eds.) *Categories, Polymorphism and Unification*, CCS - University of Edinburgh & ITLI - University of Amsterdam.

Carcagno (1989) *Text Planning for Generation of Weather Forecasts*, Technical Report, Odyssey Research Associates, Montreal.

Carcagno & Iordanskaja (1989) *Content Determination and Text Structuring in GOSSIP*, Extended Abstracts of the Second European Workshop on Natural Language Generation, Edinburgh.

Dale, Mellish & Zock (eds.) (1990) *Current Research in Natural Language Generation*, London: Academic Press.

Gardent, Bès, Jurie & Baschung (1990) Efficent Parsing for French, in *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, Univ. of British Columbia, Vancouver, 26-29 June 1989, pp.280-287.

Gardent & Plainfossé (1990) Generating from a Deep Structure, in *Proceedings of the 13th International Conference on Computational Linguistics*, Vol.2, Univ. of Helsinki, Helsinki, 20-25 August 1990, pp.127-132.

Kohl, Plainfossé & Gardent (1990) The General Architecture of Generation in ACORD, in *Proceedings of the 13th International Conference on Computational Linguistics*, Vol.3, Univ. of Helsinki, Helsinki, 20-25 August 1990, pp.388-390.

Kohl, Plainfossé, Reape & Gardent (1989) Text Generation from Semantic Representation, Deliverable T2.10, ESPRIT Project 393 ACORD, Edinburgh.

Gleitman (1965) Coordinating Conjunctions in English, *Language* 41, pp.260-293.

Halliday (1985) *An Introduction to Functional Grammar*, London: Edward Arnold.

Hovy (1989) *Notes on Dialogue Management and Text Planning in the LILOG Project.* Unpublished working document, Project LILOG, Institut für Wissenbasierte Systeme, IBM Deutschland, Stuttgart, May 1989.

Hovy (1990) *Unresolved Issues in Paragraph Planning*, in Dale, Mellish & Zock (eds.) (1990).

Kamp (1981) *A Theory of Truth and Semantic Representation*, in Groenendijk, Janssen & Stokhof (eds.) *Formal Methods in the Study of Language*, Vol. 1, pp.227-322, Amsterdam: Mathematical Centre Tracts.

Karttunen (1987) *Radical Lexicalism*, Paper presented at the Conference on Alternative Conceptions of Phrase Structure, July 1986, New-York Artificial Intelligence Center, SRI International, Center for the Study of Language and Information.

Kay (1984) Functional Unification Grammar: a Formalism for Machine Translation, *Proceedings of the 10th International Conference on Computational Linguistics and the 22nd Annual Meeting of the Association for Computational Linguistics*, Stanford University, Stanford, Ca., 2-6 July 1984, pp.75-78.

Kempen (1987) A Framework for Incremental Syntactic Tree Formation, *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, 23-28 August 1987, pp.655-660.

Lakoff (1971) *If's, And's, ans But's about conjunction*, in Fillmore & Langendoen (eds.) *Studies in Linguistic Semantics*, pp.114-149, New-York: Holt, Rinehart and Winston.

Mann (1984) Discourse Structures for Text Generation, *Proceedings of Coling 1984*.

McDonald (1980) *Natural Language Production as a Process of Decision Making under Constraint*, PhD thesis, Cambridge, Mass., MIT Press.

McKeown & Paris (1987) Functional Unification Grammar Revisited, in *Proc. 25th Ann. Meet. Assoc. Comput. Ling.*, Stanford, Ca., July.

Mel'cuk (1988) *Paraphrase et lexique dans la théorie linguistique Sens-Texte*, Lexique 6, "Lexique et paraphrase", P. U. Lille, pp.13-54.

Mel'cuk et al. (1984) *Dictionnaire Explicatif et Combinatoire du Français Contemporain*, Recherches Lexico-Sémantiques I, Presses de l'Université de Montréal, Montréal, Canada.

Mel'cuk & Polguère (1987) *A Formal Lexicon in Meaning-text Theory (Or How to do Lexica with Words)*, Computational Linguistics vol.13, 3-4, pp.261-275.

Moens, Calder, Klein, Reape, Zeevat (1989) Expressing Generalizations in Unification-based Grammar Formalisms, *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, UMIST, Manchester, 10-12 April 1989, pp.174-181.

Pereira & Warren (1980) Definite Clause Grammars for language analysis - a survey of the formalism and a comparison with Augmented Transition Grammars, *Artificial Intelligence*, 13, 231-278.

Polguère (1990) *Sructuration et mise en jeu procédurale d'un modèle linguistique déclaratif dans un cadre de génération de texte*, PhD dissertation, University of Montréal.

Richtie (1986) The Computational Complexity of Sentence Derivation in Functional Unification Grammar, *Proceedings 11th Int. Conf. Comput. Ling.*, Bonn, W. Germany, August, pp.584-586.

Shieber (1986) *An Introduction to Unification-based Approaches to Grammar*, CSLI Lecture Notes 4, Stanford.

Shieber (1988) An Uniform Architecture for Parsing and Generation, in *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, 22-27 August 1988, pp.614-619.

Shieber, Uszkoreit, Pereira, Robinson & Tyson (1983) The Formalism and Implementation of PATR-II, in Grosz & Stickel (eds.) *Research on Interactive Acquisition and Use of Knowledge*, SRI International, Menlo Park, pp.39-79.

Shieber, van Noord, Moore & Pereira (1989) A Semantic Head-driven Generation Algorithm for Unification-based Formalisms, in *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, Univ. of British Columbia, Vancouver, 26-29 June 1989, pp.7-17.

van Noord (1989) *BUG: a directed Bottom-Up Generator for unification-based formalisms*, Ms., Department of Linguistics, University of Utrecht.

Zeevat (1986) *A specification of InL*, Internal ACORD Report, Centre for Cognitive Science, Edinburgh.

Zeevat, Klein & Calder (1987) An Introduction to Unification Categorial Grammar, in Haddock, Klein & Morrill (eds.) *Edinburgh Working Papers in Cognitive Science, Vol.1: Categorial Grammar, Unification Grammar and Parsing*, Centre for Cognitive Science, Edinburgh.

3.1.2 MEANING-TEXT THEORY

3.1.2.1 Approach

The Meaning-Text Theory (MTT) was put forward in 1965 by two Soviet linguists, Alexander Zholkovsky and Igor' Mel'cuk, later joined by Jurij Apresjan. MTT was conceived and developed as a general theory of human language. MTT views a natural language as a logical device which establishes the correspondence between the infinite set of all possible meanings and the infinite set of all possible texts and vice versa. This device ensures the construction of linguistic utterances which express a given meaning and the comprehension of possible meanings expressed by a given utterance. This device can be seen as a cybernetic model with a system of rules approximating the Meaning <=> Text correspondence.

3.1.2.1.1 The seven levels

The MTM has to match a given meaning with many different texts and a great many different texts have to be reduced to the same meaning representation. This makes it almost impossible to establish a direct correspondence between semantic and phonological representation, and so two intermediary levels of utterance representation have to be introduced, syntactic and morphological, the former aimed at the sentence as a structural object and the latter dealing with the word. All levels, except for the semantic, are split into two sub-levels: a deep one, geared to meaning, and a surface one, determined by physical form. This gives a total of seven representation levels:

1 - Semantic Representation (SemR);
2 - Deep Syntactic Representation (DSyntR);
3 - Surface Syntactic Representation (SSyntR);
4 - Deep Morphological Representation (DMorphR);
5 - Surface Morphological Representation (SMorphR);
6 - Deep Phonetic Representation (DPhonR);
7 - Surface Phonetic Representation (SPhonR).

A representation is a set of formal objects called **structures**, with one considered as the main one and all the others specifying some of its characteristics. Each structure depicts a certain aspect of the item considered at a given level.

3.1.2.1.2 The six components

A MTM has the task of establishing correspondences between the semantic representation and the morphological (written utterance) or phonetic (spoken utterance) through the intermediate levels. Accordingly, the MTM consists of the following six basic components:

1 - the semantic component or semantics
2 - the deep syntactic component or deep syntax
3 - the surface syntactic component or surface syntax
4 - the deep morphological component or deep morphology
5 - the surface morphological component or surface morphology
6 - the deep phonetic component or phonemics.

3.1.2.1.3 The explanatory-combinatorial dictionary

The Meaning-Text Theory puts strong emphasis on the development of highly structured lexica. It assigns to the lexicon a central place, so that therest of linguistic description is supposed to pivot around the lexicon. We will present here such a lexicon, the Explanatory Combinatorial Dictionary (ECD), developed within the framework of MTT.

A lexicographic unit in the ECD, i.e. a dictionary entry, covers one lexical item taken in one well-specified sense. All such items called lexemes are described in a rigourous and uniform way, so that a dictionary entry is divided into three major zones: the semantic zone, the syntactic zone, and the lexical co-occurrence zone.

1 The semantic zone

The semantic definition is a decomposition of the meaning of the corresponding lexeme. It is a semantic network whose nodes are labeled either with semantic units (lexemes) or with variables, and whose arcs are labeled with distinctive numbers which identify different arguments of a predicate. A lexical label represents the definition (the meaning) of the corresponding lexeme, rather than the lexeme itself. Therefore, each node of a definitional network stands, in its turn, for another network, whose nodes are replaceable by their corresponding networks, and so forth, until the bottom level primitives are reached.

2 The syntactic zone

This zone stores the data on the syntactic behaviour of the head lexeme. Along with the part of speech (syntactic category), the syntactic zone presents two major types of information:

- Syntactic features
- The government pattern.

a - Syntactic features

A syntactic feature of a lexeme specifies particular syntactic structures which accept it but which are not directly related to the semantic actants appearing in its definition. Syntactic features, which do not presuppose strictly disjoint sets, provide for a more flexible and multi-faceted sub-classification of lexemes than do parts of speech, which induce a strict partition of the lexical stock.

b - The government pattern

The government pattern of a lexeme specifies the correspondence between its semantic actants and their realization at the DSynt-level and DMorph-level. It is a rectangular matrix with three rows:

- The upper one contains semantic actants of the lexeme;
- The middle one indicates the DSynt-roles played by the manifestations of the semantic actants on the DSynt-level;
- The lower one indicates structural words and morphological forms necessary for the manifestation of the same semantic actants on the SSynt- and DMorph- levels.

The number of columns in this matrix is equal to the number of semantic actants. Each column specifies the correspondence between a semantic actant and its realization on closer-to-surface levels. In general, a government pattern has associated with it a number of restrictions concerning the co-occurrence and the realization of actants:

- An actant cannot appear together with/without another actant;
- A given surface form of an actant determines the surface form of another actant;
- A given realization of an actant is possible only under given conditions, semantic or otherwise.

These restrictions function as filters screening possible forms and combinations of actants on the DSynt- as well as on the SSynt-level.

c - The lexical zone

The main novelty of the ECD is a systematic description of the restricted lexical co-occurrence of every head lexeme. This description uses lexical functions.

A lexical function is a dependency that associates with a lexeme another lexeme (or a set of synonymous lexemes) which expresses a very abstract meaning and plays a specific syntactic role.

For instance, for a noun N denoting an action, the lexical function Oper1 specifies a verb (semantically empty or at least emptied) which takes as its grammatical subject the mame of the agent of the said action and as its direct object the lexeme N itself.

Oper1(QUESTION) = ASK
Oper1(QUESTION) = POSER
Oper1(PREGUNTA) = HACER
Oper1(VOPROS) = ZADAT'

There are about 60 lexical functions of the Oper1 type, called standard elementary LFs. They and their combinations allow one to describe exhaustively and in a highly systematic way almost the whole of restricted lexical co-occurrence in natural languages.

In MTM lexical functions play a double role:

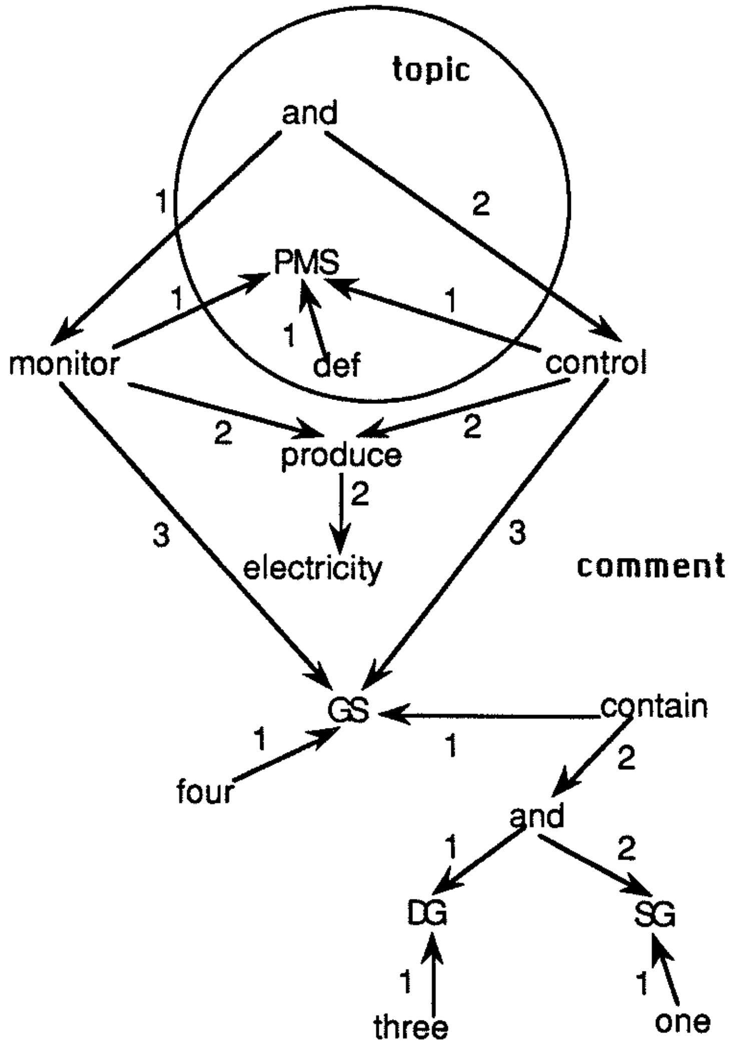
1 - During the production of the text from a given SemR, LFs control the proper choice of lexical items linked to the lexeme by regular semantic relations. During the analysis of a text, LFs help to resolve syntactic homonymy, since they indicate which word has the greater likelihood of going with which other word.

2 - In text production, LFs are used to describe sentence synonymy, or more precisely, the derivation of a set of synonymous sentences from the same DSyntS. This is done by formulating, in terms of LFs, a number of equivalences. The operation carried out by these rules is called paraphrasing. About sixty paraphrasing rules are needed to cover all systematic paraphrases in any language. Moreover there must be about thirty syntactic rules which describe transformations of trees and serve the rules of the first type.A powerful paraphrasing system is necessary, not only because it is interesting in itself, but mainly because without such a system it seems impossible to produce texts of good quality for a given SemR. When one is blocked during a derivation by linguistic restrictions, one can by-pass the obstacle by recourse to paraphrases. During text analysis, a powerful paraphrasing system helps to reduce the vast synonymy of natural language to a standard and therefore more manageable representation.

3.1.2.2 The Meaning-Text Model

3.1.2.2.1 The semantic representation

The SemR of an utterance consists of two structures: the semantic structure(SemS) and the semantic-communicative structure (SCommS).



1 The semantic structure

The semantic structure specifies the meaning of the utterance independent of its linguistic form. The distribution of meaning among words, clauses, or sentences is ignored; so are such linguistic features as the selection of specific syntactic constructions and so on. At the same time, the SemS tries to depict the meaning objectively, leaving out the speaker and his intentions, which are taken into account in the second structure of the SemR. Formally, a SemS is a connected graph or a network.

The vertices or nodes of a SemS are labeled with semantic units, or **semantemes**. Two major classes of semantemes are distinguished:

1- **functors**, further subdivided into **predicates** (relations, properties, actions, states, events); **logical connectives** (*if, and, or, not*) and **quantifiers** (*all, there exist,* numbers);

2 - **names (of classes) of objects**, including proper names.

Both types of semantemes can receive arcs or arrows, but only a functor can head an arrow. The arrows on the arcs point from functors to their arguments.

The arcs of a SemS are labeled with numbers which have no meaning of their own but only serve to differentiate the various arguments of the same functor.

2 The communicative structure

The SCommS specifies the intentions of the speaker with respect to the organization of the message. The same meaning reflecting a given situation can be encoded in different messages according to what the speaker wants. The SCommS must show at least the following contrasts:

a - Theme (topic) vs rheme (comment), i.e. the starting point of the utterance, its source, as opposed to what is communicated about the topic.
b - Old, or given (known to both interlocutors) vs new, i.e. communicated by the speaker.
c - Foregrounded (expressed as a main predication) vs backgrounded (relegated to an attribute).
d - Emphatically stressed vs neutral.

However, a new model has been designed during the Exploratory Action (cf Task 2.2), which realizes a better account for communicative aspects.

3 The semantic component

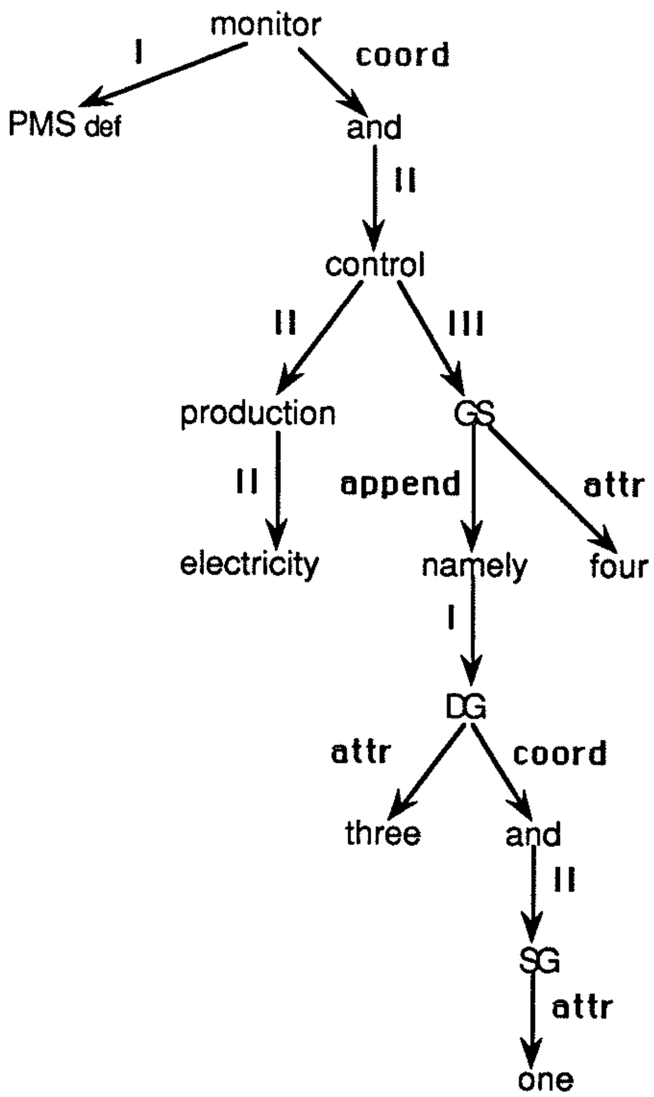
It establishes the correspondence between the SemR of an utterance and all the synonymous sequences of DSyntRs of the sentences that make up that utterance. To do that, it performs the following operations:

- It selects the corresponding lexemes by means of semantic-lexical rules.
- It supplies meaning-bearing morphological values of lexemes by means of semantico-morphological rules.
- It forms a tree out of the lexemes it has chosen.
- It introduces the anaphoric structure, that is, it indicates coreferences for the lexical nodes that have appeared as a result of the duplication of some semantic nodes.
- It computes the prosody of the sentence on the basis of semantico-prosodic rules.
- It provides the communicative structure from the data contained in the SCommS.
- For each DSyntR produced, the semantic component constructs all the synonymous DSyntRs that can be exhaustively described in terms of lexical functions. This is achieved by means of a paraphrasing system that defines an algebra of transformations on such DSyntRs where the DSyntS contains symbols lexical functions.

3.1.2.2.2 The deep syntactic representation

A DSyntR consists of four structures: the deep syntactic structure (DSyntS), the deep syntactico-communicative structure, the deep syntactico-anaphoric structure, the deep syntactico-prosodic structure.

1 The deep syntactic structure



The DSyntS is a dependency tree which represents the syntactic organization of the sentence in terms of its constituent words and relationships between them.

A node of a DSyntS is labeled with a generalized lexeme of the language. A generalized lexeme is:

1 - a full lexeme of the language (semantically empty words, like governed prepositions and conjunctions or auxiliary verbs are left out);
2 - an idiom;
3 - a lexical function.

A branch of a DSyntS is labeled with the name of a deep syntactic relation. There are nine relations:

- ,II,...,VI are six predicative relations connecting a semantically predicative lexeme with its 1st, 2nd,..., 6th arguments, respectively;

- ATTR is the attributive relation, which covers all kinds of modifiers and attributes (in the broad sense);
- COORD is a relation that accounts for all coordinate or conjoined constructions;
- APPEND is an appendancy relation that subsumes all parentheticals, interjections, addresses, linking any of these elements to the top node (main verb) of the corresponding clause.

There is no linear order of nodes within the DSyntS. Word order is taken to be a means for encoding syntactic structure into speech strings and therefore it is banned from the syntactic structure.

The communicative structure is close to the one of the semantic level.
The anaphoric structure carries the information about coreferentiality.
The prosodic structure represents intonation contours, pauses, emphatic stresses.

2 The deep syntactic component

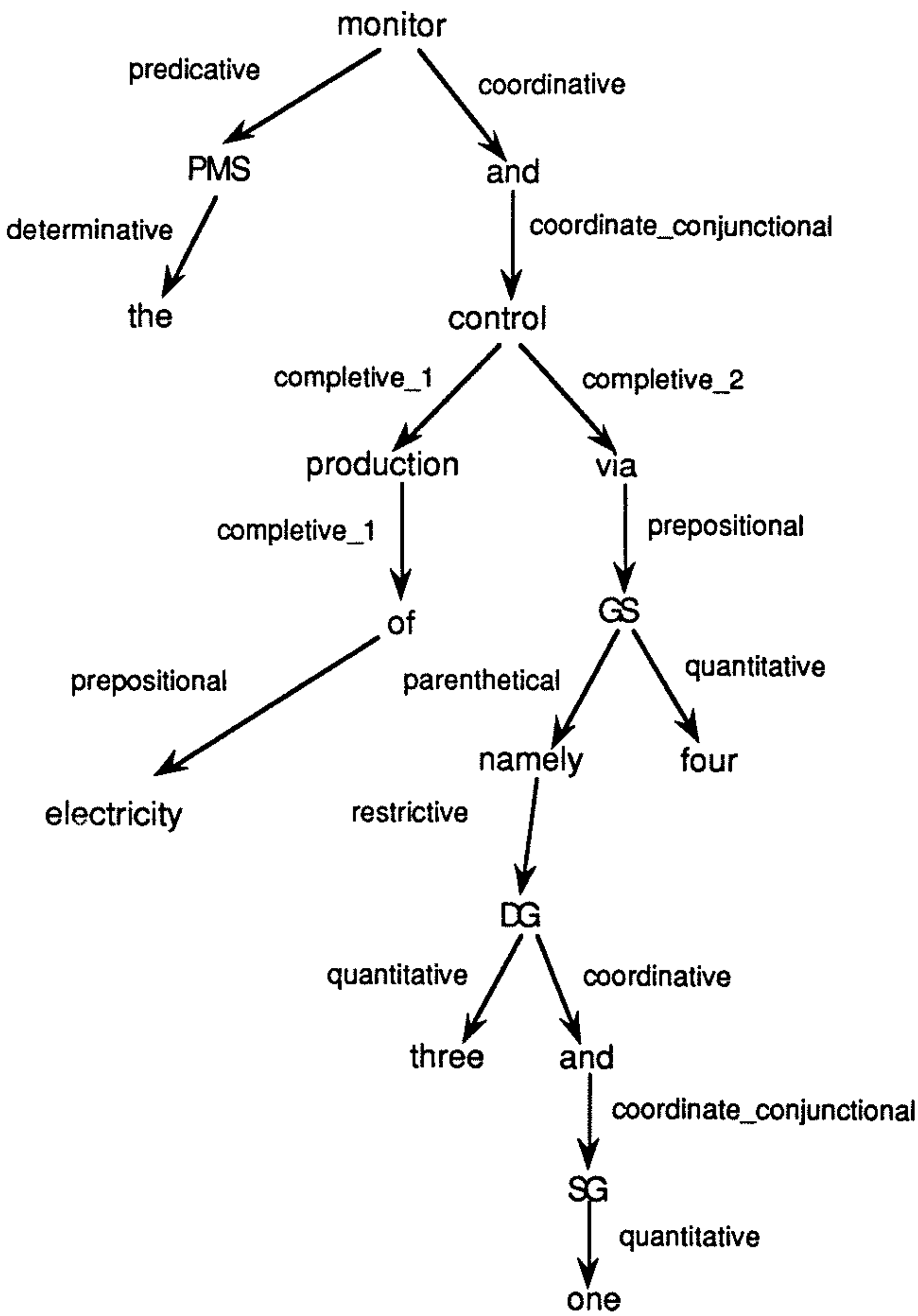
It establishes the correspondence between the DSyntR of a sentence and all the alternative SSyntRs which correspond to it. To do that, it performs the following operations:

1 - It computes the values of all lexical functions.
2 - It expands the nodes of idioms into corresponding surface trees.
3 - It eliminates some nodes that occur in anaphoric relations and should not appear in actual text.
4 - It constructs the SSyntS by means of transformations.
5 - It processes the three other structures of the SSyntR.

3.1.2.2.3 The surface syntactic representation

It consists of four structures corresponding to those of the DSyntR. The SSyntS is also a dependency tree but its composition and labeling differ sharply from those of the DSyntS.

1 The surface syntactic structure



A node is labeled with an actual lexeme of the language. First, all the lexemes are represented, including the semantically empty ones. Second, all the idioms are expanded into actual surface trees. Third, the values of all the lexical functions are computed (on the basis of the lexicon). Fourth, all pronominal replacements and deletions under lexical or referential identity are carried out.

A branch is labeled with the name of a surface syntactic relation. A relation belongs to a set of language-specific binary relations, each describing a particular syntactic construction.

As is the case with the DSyntS, the nodes of the SSyntS are not ordered linearly. This enables us to keep strictly apart two basically different "orders": syntactic hierarchy and linear ordering, which serves to express this hierarchy.

The communicative structure, the anaphoric structure and the prosodic structure are analogous to their deep counterparts.

2 The surface syntactic component

It establishes the correspondence between the SSyntR of a sentence and all the alternative DMorphRs that are realizations of it. It performs the following operations:

1 - Morphologization of the SSyntS: it determines all the syntactically conditioned morphological values of all the words, such as the number and person of the verb.
2 - Linearization of the SSyntS: it determines the actual word order of the sentence.
3 - Ellipsis: it carries out all kinds of conjunction reductions and deletions that are prescribed by the language.
4 - Punctuation: it determines, on the basis of the prosodic structure, as well as on the basis of the resulting SSyntS, the correct prosody which, in the case of printed text, is rendered by punctuation.

The basic tool of morphologization and linearization is the syntagm or SSynt-rule. Beside syntagms, surface syntax uses four additional types of rules:

- Word order patterns for elementary phrases
- Global word order patterns
- Ellipsis rules
- Prosodic or punctuation rules.

3.1.2.2.4 The deep morphological representation

It is a string of representations of all the wordforms that compose the sentence.

3.1.2.3 References

Mel'cuk, I.A. (1981) "Meaning-Text Models: A Recent Trend in Soviet Linguistics", in *Annual Review of Anthropology*, 10.

Mel'cuk, I.A. (1988) *Dependency Syntax: Theory and Practice*, New York, SUNY Press.

Mel'cuk, I.A. & Polguere A. (1987) "A Formal Lexicon in the Meaning-Text Theory", in *Computational Linguistics*, 13.

3.2 Evaluation of models according to results of corpus analysis

3.2.1 - UCG

The survey of existing documentation (cf. 2.2) has lead to the conclusion that its main linguistic feature is *simplicity*.

The salient *lexical/syntactical* phenomena are nominalizations, anteposition of subordinate clauses and extensive use of passives - which can easily be handled within UCG. To sum up, there are no (or nearly no) converses, topicalization, cleft sentences, verb-subject inversions, impersonal forms, few relatives, completives, infinitives, PP modifiers, and only some adverbials.

With respect to *semantics*, the restricted phenomena related to quantification and determination, pronominalisation, comparison, substitution and ellipsis are obviously not beyond the scope of UCG. An adequate treatment of the phenomena to be emphasized (i.e. coordination and nominal composition) seems quite feasible, according to the fact that UCG embodies DRT and may thus rely on its formal properties.
Moreover, the absence of interrogatives, unbounded dependencies, etc., makes the task of *linearization* fairly simple, inasmuch as the ordering within the sentences is almost always canonical.

3.2.2 - Meaning-Text Theory

Here we will examine how MTT tackles the problems encountered in the corpus analysis considering:

- communicative organization
    - lexicon
    - determination
    - syntax
- cohesion
    - reference and coreference
    - substitution and ellipsis
    - coordination and subordination.

<u>3.2.2.1 - Communicative organization</u>

3.2.2.1.1 - Lexicon

Choices, lexical or syntactic, are either determined by communicative and cohesive constraints or freely made as a paraphrasing device. In standard MTT lexical functions are used in both cases at the deep-syntactic level. In GLOSE it is only when using the paraphrasing power of the model that the lexical functions are triggered as such; otherwise, when seen as a device for fulfilling a given communicative goal, they are used as a means of producing the right lexemes in a straightforward way with no intermediate lexicalization.

Whether linked to the issues of communicative organization, distribution of information or cohesion, semantic relations between lexical items play a major role.Semantic relations such as converse, synonymy, hyperonymy, antonymy, typical actants are not much used in the present corpus but yet they are an important device for communicative organization and cohesion. Therefore the generation model should be able to express such relations.

In MTT these relations are to be found in the dictionary and are described by lexical functions.

Lexical functions are used at the deep syntactic level as a means of paraphrasing. This is due to the fact that in order to use lexical functions a meaning has to be lexicalized and only then can the dictionary be accessed and lexical functions triggered. But this also means that the communicative organization is not taken into account. Yet it is the communicative organization which, among others, imposes constraints on the choice of the lexemes. Therefore these constraints should be dealt with at the semantic level, prior to and not after lexicalization.

For instance the action of buying can be expressed either by *buy* or *sell*, possession either by verbs such as have or *possess* or by the genitive case, causality either by the verb *cause* or by a subordinate clause introduced by *if.*

Buy and *sell* are traditionally considered as converses. They belong to the same syntactic category. As for the relations of possession and causality they are not expressed by synonyms belonging to the same syntactic category.

In MTT
- the lexical function Syn can be used for synonyms belonging to the same syntactic category such as *have* and *possess*;
- the lexical function Conv can be used for converses belonging to the same syntactic category such as *buy* and *sell*;
- there are lexical functions for derived terms such as *possess/possession*;
- BUT there is no lexical function for pairs of terms such as *if/cause*.

The solution adopted in GLOSE is to consider the semantic level as a "notional" level made of nodes which will be lexicalized taking all communicative constraints into account from the start and not going via an intermediate level where nodes are first lexicalized so as to later access the right lexemes through lexical functions.

3.2.2.1.2 - Determination

In MTT determination is poorly represented. At the semantic level there are two nodes 'def' and 'more than one' supposed to indicate the definite and plural character of the determined node. The absence of 'def' for instance indicates that an indefinite article should be produced. The problem is that these indications are not sufficient: the definiteness and the plurality of a node do not provide enough information for choosing the right determiner. Besides, 'def' is not a primitive value. A determiner may be definite for many different reasons. The semantic values of determiners are much more complex: generic vs specific vs non-specific, distributive vs collective, etc.

In GLOSE a distinction is made between determination and quantification.Information concerning determination is represented as features attached to the determined node and information concerning quantification is represented as a functor with predicate and quantified NP as arguments.

Determination concerns
- plurality
- generic/specific value

and quantification is either

- determinate (*one, two, all*)
- or indeterminate (*some, several*).

If we consider determiners as a means of expressing the difference between given and new information, we should also take these values into account and study their interaction with the other semantic values we mentioned. To Mel'cuk the given/new distinction is part of the communicative structure.

3.2.2.1.3 - Syntax

Syntactic constructions are either constrained by a given communicative orientation or seen as a paraphrasing tool.

The syntactic constructions in the corpus were the following:

- active vs passive
- inversion verb-subject
- impersonal form
- relatives
- completive
- infinitive
- nominalization
- subordinate

These constructions may be generated at different levels through various devices:

- <u>active vs passive</u>: this choice is either made during the transition from RSem to RSP if constrained or made later if seen as paraphrasing;

- <u>inversion verb-subject:</u> this choice is made during the last transition, from RSS to RMorph, since in MTT there is no earlier indication for linearization;

- <u>impersonal form:</u> this form is either constrained by the absence of first actants at the semantic level or chosen according to the government pattern of the lexeme;

- <u>relatives</u>: there should be a different representation for descriptive vs restrictive relatives;

- <u>completive</u>: indicated in the government pattern of "mother" lexeme;

- <u>infinitive</u>: indicated in the government pattern of "mother" lexeme;

- <u>nominalization</u>: either constrained by communicative orientation and realized during the transition from RSem to RSP or constrained by the government pattern of its "mother" lexeme;

- <u>subordinate</u>: at the semantic level the relation holding between the future clauses is represented and is lexicalized during the transition toward RSP according to the communicative orientation. The ordering of the two clauses will be carried out during the transition from RSS toRMorph.

3.2.2.2 - Cohesion

3.2.2.2.1 - Reference and coreference

In MTT anaphoric links are created during the transition from RSem to RSP; at the RSS level pronominalization is carried out.

Reference is not properly treated in MTT. In GLOSE [Gobinet 90] referential blocks were introduced so as to have a better representation of reference. Therefore the approach of coreference in GLOSE is more restricted than the one in MTT. As for determination a distinction is made as mentioned above between determination proper and quantification.

All this contributes to a more constrained and controlled paraphrasing system than MTT. The idea is not to generate all possible paraphrases but to constrain the paraphrasing power in order to produce the right sentence in a given context.

3.2.2.2.2 - Substitution and ellipsis

Substitution and ellipsis are procedures applied at different levels in MTM depending on the nature of these operations. Anyway the RSem at the start must be a full RSem.

- Substitution is in most cases decided during the transition from RSem to RSP: a whole sub-network can be represented by a single general lexeme as *do* for verbal substitution, *one* for nominal and *so* for clausal.

- Ellipsis is most often carried out during the transition from RSS to RMorph. In RSS the node which will be omitted is still present. For instance, in "*John kissed Ann and Bob kissed Mary*", the second "*kissed*" will be elided during linearization, and the result will be "*John kissed Ann and Bob Mary*".

3.2.2.2.3 - Coordination and subordination

- Coordination

In the corpus there is an extensive use of coordination, with mostly *and* and *or*. Coordination can be a problem in the dependency approach.Mel'cuk mentions this problem and suggests some solutions [DS].

To Mel'cuk there can be a symmetry at the semantic level ("*John and Mary*" is identical to "*Mary and John*") but not at the syntactic level. "In the majority of cases there is no reversibility in coordinated structures".

(1a) *He stood up and gave me a letter.*
(1b) *He gave me a letter and stood up.*

(2a) *Go to bed or I'll spank you!*
(2b) *I'll spank you or go to bed!*

From a purely syntactic point of view, the left conjunct and the conjunction phrase are not equal: there is a dependency relation between them. The conjunction phrase depends syntactically on the left conjunct. Within the conjunction phrase itself, the conjunct introduced by the conjunction depends on it: *John -> and -> Mary* or *stood up -> and -> gave.*

Dependency trees in MTT cannot express the difference which lies:

- when the modification of the head of the phrase by an element contrasts with the modification of the whole phrase by the same element: for instance, in French, "*sa gaiete etonnante et son accent*" and "*sa gaiete et son accent etonnants*";

- or when the modification of an element X by a phrase contrasts with the modification of X by separate elements of the same phrase: for instance, in English, "*Bob and Dick's novels*" and "*Bob's and Dick's novels*".

Mel'cuk suggested several possible solutions:

- have two different labels : "modif" and "phrase-modif";
- consider these forms as ellipsis;
- consider the differences as meaningful and retain them in the syntactic structure;
- introduce groupings: this solution is probably the best and can be adopted in some contexts where dependency-language proves insufficient.

A grouping is not like a phrase-structure constituent: its elements are not linearly ordered, dependency relations are explicit, and there is no higher node to represent the grouping as a whole. This grouping device can be used for conjoined structures and structures in which the syntactic scope of "operators" (negation, *only*) plays a role.

- Subordination

There are not many different subordinate conjunctions (mostly *when* and *if*) in the present corpus but they are widely used. Besides in case we want to use conjunctions as a cohesive device, we need a satisfying representation of subordinating conjunctions. In GLOSE semantic relations holding between clauses are represented at the semantic level as predicates. The arguments corresponding to the clauses are considered as "propositional blocks". The semantic relations will be lexicalized, depending on communicative orientation, as lexemes belonging to different syntactic categories.

- Communicative organization
  - lexicon
  - determination
  - syntax

- Cohesion
  - reference and coreference
  - substitution and ellipsis
  - coordination and subordination.

3.2.2.1 Communicative organization

3.2.2.1.1 Lexicon

Choices, lexical or syntactic, are either determined by communicative and cohesive constraints or freely made as a paraphrasing device. In standard MTT lexical functions are used in both cases at the deep-syntactic level. In GLOSE it is only when using the paraphrasing power of the model that the lexical functions are triggered as such; otherwise, when seen as a device for fulfilling a given communicative goal, they are used as a means of producing the right lexemes in a straightforward way with no intermediate lexicalization.

Whether linked to the issues of communicative organization, distribution of information or cohesion, semantic relations between lexical items play a major role.Semantic relations such as converse, synonymy, hyperonymy, antonymy, typical actants are not much used in the present corpus but yet they are an important device for communicative organization and cohesion. Therefore the generation model should be able to express such relations.

In MTT these relations are to be found in the dictionary and are described by lexical functions.

Lexical functions are used at the deep syntactic level as a means of paraphrasing. This is due to the fact that in order to use lexical functions a meaning has to be lexicalized and only then can the dictionary be accessed and lexical functions triggered. But this also means that the communicative organization is not taken into account. Yet it is the communicative organization which, among others, imposes constraints on the choice of the lexemes. Therefore these constraints should be dealt with at the semantic level, prior to and not after lexicalization.

For instance the action of buying can be expressed either by *buy* or *sell*, possession either by verbs such as have or *possess* or by the genitive case, causality either by the verb *cause* or by a subordinate clause introduced by *if*.

Buy and *sell* are traditionally considered as converses. They belong to the same syntactic category. As for the relations of possession and causality they are not expressed by synonyms belonging to the same syntactic category.

In MTT

- the lexical function Syn can be used for synonyms belonging to the same syntactic category such as *have* and *possess*;
- the lexical function Conv can be used for converses belonging to the same syntactic category such as *buy* and *sell*;
- there are lexical functions for derived terms such as *possess/possession*;
- BUT there is no lexical function for pairs of terms such as *if/cause*.

The solution adopted in GLOSE is to consider the semantic level as a "notional" level made of nodes which will be lexicalized taking all communicative constraints into account from the start and not going via an intermediate level where nodes are first lexicalized so as to later access the right lexemes through lexical functions.

3.2.2.1.2 Determination

In MTT determination is poorly represented. At the semantic level there are two nodes 'def' and 'more than one' supposed to indicate the definite and plural character of the determined node. The absence of 'def' for instance indicates that an indefinite article should be produced. The problem is that these indications are not sufficient: the definiteness and the plurality of a node do not provide enough information for choosing the right determiner. Besides, 'def' is not a primitive value. A determiner may be definite for many different reasons. The semantic values of determiners are much more complex: generic vs specific vs non-specific, distributive vs collective, etc.

In GLOSE a distinction is made between determination and quantification.Information concerning determination is represented as features attached to the determined node and information concerning quantification is represented as a functor with predicate and quantified NP as arguments.

Determination concerns

- plurality
- generic/specific value

and quantification is either

- determinate (*one, two, all*)
- or indeterminate (*some, several*).

If we consider determiners as a means of expressing the difference between given and new information, we should also take these values into account and study their interaction with the other semantic values we mentioned. To Mel'cuk the given/new distinction is part of the communicative structure.

3.2.2.1.3 Syntax

Syntactic constructions are either constrained by a given communicative orientation or seen as a paraphrasing tool.

The syntactic constructions in the corpus were the following:

- Active vs passive
- Inversion verb-subject
- Impersonal form
- Relatives
- Completive
- Infinitive
- Nominalization
- Subordinate

These constructions may be generated at different levels through various devices:

- Active vs passive: this choice is either made during the transition from RSem to RSP if constrained or made later if seen as paraphrasing;
- Inversion verb-subject: this choice is made during the last transition, from RSS to RMorph, since in MTT there is no earlier indication for linearization;
- Impersonal form: this form is either constrained by the absence of first actants at the semantic level or chosen according to the government pattern of the lexeme;
- Relatives: there should be a different representation for descriptive vs restrictive relatives;
- Completive: indicated in the government pattern of "mother" lexeme;
- Infinitive: indicated in the government pattern of "mother" lexeme;

- Nominalization: either constrained by communicative orientation and realized during the transition from RSem to RSP or constrained by the government pattern of its "mother" lexeme;

- Subordinate: at the semantic level the relation holding between the future clauses is represented and is lexicalized during the transition toward RSP according to the communicative orientation. The ordering of the two clauses will be carried out during the transition from RSS toRMorph.

3.2.2.2 Cohesion

3.2.2.2.1 Reference and coreference

In MTT anaphoric links are created during the transition from RSem to RSP; at the RSS level pronominalization is carried out.

Reference is not properly treated in MTT. In GLOSE [Gobinet 90] referential blocks were introduced so as to have a better representation of reference. Therefore the approach of coreference in GLOSE is more restricted than the one in MTT. As for determination a distinction is made as mentioned above between determination proper and quantification.

All this contributes to a more constrained and controlled paraphrasing system than MTT. The idea is not to generate all possible paraphrases but to constrain the paraphrasing power in order to produce the right sentence in a given context.

3.2.2.2.2 Substitution and ellipsis

Substitution and ellipsis are procedures applied at different levels in MTM depending on the nature of these operations. Anyway the RSem at the start must be a full RSem.

- Substitution is in most cases decided during the transition from RSem to RSP: a whole sub-network can be represented by a single general lexeme as *do* for verbal substitution, *one* for nominal and *so* for clausal.

- Ellipsis is most often carried out during the transition from RSS to RMorph. In RSS the node which will be omitted is still present. For instance, in "*John kissed Ann and Bob kissed Mary*", the second "*kissed*" will be elided during linearization, and the result will be "*John kissed Ann and Bob Mary*".

3.2.2.2.3 Coordination and subordination

Coordination

In the corpus there is an extensive use of coordination, with mostly *and* and *or*. Coordination can be a problem in the dependency approach.Mel'cuk mentions this problem and suggests some solutions [DS].

To Mel'cuk there can be a symmetry at the semantic level ("*John and Mary*" is identical to "*Mary and John*") but not at the syntactic level. "In the majority of cases there is no reversibility in coordinated structures".

(1a) *He stood up and gave me a letter.*
(1b) *He gave me a letter and stood up.*

(2a) *Go to bed or I'll spank you!*
(2b) *I'll spank you or go to bed!*

From a purely syntactic point of view, the left conjunct and the conjunction phrase are not equal: there is a dependency relation between them. The conjunction phrase depends syntactically on the left conjunct. Within the conjunction phrase itself, the conjunct introduced by the conjunction depends on it: *John -> and -> Mary* or *stood up -> and -> gave.*

Dependency trees in MTT cannot express the difference which lies:

- When the modification of the head of the phrase by an element contrasts with the modification of the whole phrase by the same element: for instance, in French, "*sa gaiete etonnante et son accent*" and "*sa gaiete et son accent etonnants*";

- Or when the modification of an element X by a phrase contrasts with the modification of X by separate elements of the same phrase: for instance, in English, "*Bob and Dick's novels*" and "*Bob's and Dick's novels*".

Mel'cuk suggested several possible solutions:

- Have two different labels : "modif" and "phrase-modif";
- Consider these forms as ellipsis;
- Consider the differences as meaningful and retain them in the syntactic structure;
- Introduce groupings: this solution is probably the best and can be adopted in some contexts where dependency-language proves insufficient.

A grouping is not like a phrase-structure constituent: its elements are not linearly ordered, dependency relations are explicit, and there is no higher node to represent the grouping as a whole. This grouping device can be used for conjoined structures and structures in which the syntactic scope of "operators" (negation, *only*) plays a role.

Subordination

There are not many different subordinate conjunctions (mostly *when* and *if*) in the present corpus but they are widely used. Besides in case we want to use conjunctions as a cohesive device, we need a satisfying representation of subordinating conjunctions. In GLOSE semantic relations holding between clauses are represented at the semantic level as predicates. The arguments corresponding to the clauses are considered as "propositional blocks". The semantic relations will be lexicalized, depending on communicative orientation, as lexemes belonging to different syntactic categories.

3.3 Evaluation of models according to general criteria

3.3.1 Identification of criteria for evaluating text generation systems

Experience in development of several implementations of text generation systems has led us to specifications of general criteria which help to compare different linguistic frameworks. These general criteria thus take into account current limits of the state of the art in Natural Generation systems as well as specific requirements implied by the aim of multilingual generation. The need for integrating NL generation technologies into practical contexts of application is also considered.

3.3.1.1 Advanced software engineering principles

A first general principle which should be respected by a generation component is **declarativity**. Declarativity, in our sense, means knowledge distinguished from control. A second principle is **modularity** of the system which is expressed in an architecture where functions correspond to different modules. Achieving such principles leads to a greater efficiency in development, testing, maintenance and modification steps of the lifecycle.

3.3.1.2 Linguistic engineering principles

Complexity of models describing languages and requirement of modularity impose a demanding architecture on the generation model. The model must involve a **high level of structuring of lexica and grammars.**

A complete coverage of linguistic phenomena which may be encountered in applications is very difficult to attain. Therefore, models must be measured according to the **extent of their linguistic description,** i.e. list of phenomena handled by the model, and potential ability of the model to handle new phenomena.

Models must include a **methodology of description** for lexica and grammars. This methodology will allow to achieve consistency when building incrementally these knowledge bases. It will also ensures portability of the system to new languages or sub-languages.

Models must demonstrate their **ability to deal successfully with real life applications,** taking into account significant lexicon, grammar and conceptual coverage.

3.3.1.3 Multilingual generation engineering principles

A good measure of the linguistic ability of the model is its **paraphrasis power**. A powerful generator should have the ability to synthesize many different sentences from a single meaning representation. This ability implies a flexible collaboration of concurrent knowledge sources and the management of several potential solutions. However, it is often the case that despite the possibility of conveying a given meaning in a number of different ways, there are according to the context only few (more often one) sentences that fit better than the others. Therefore, paraphrasis must be controlled and the choice must be made according to explicit linguistic knowledge instead of a blind combinatorial criterion like backtracking.

The model must be **independent relatively to a specific language**. This may be achieved through a comparative study of different languages. This may also entail a clear-cut distinction between trans-linguistic components and specific language components.

The model must show also **independence towards a specific application** by involving a general interface to the application program. Different applications such as databases, expert systems, command and control systems, CAD and CASE environments should be made available through that interface.

3.3.2 Evaluation of MTT according to the criteria above

3.3.2.1 Advanced software engineering principles

Linguistic Knowledge in MTT is described by the lexicon and by rule bases which guide transitions between the representation levels of the model. Control is performed by a separate engine which interprets transition rules according to a given representation in order to produce the next level of representation. In addition, standard procedures of network and tree traversals are used in order to analyze current representations. Thus, MTT achieves declarativity by a clear separation between Knowledge and Control. This approach makes easier the adding of new knowledge, like communicative constraints.

Modularity of treatments and stratificational structuring of linguistic representations in MTT allow to locate precisely where a given knowledge is effective and where a given decision operates.

For instance, the modularity observed when handling separately the order of constituents and their functional composition allows a better account for constituent order between different languages. This is recognized as a major asset of dependency grammars and entails an easiest treatment of linearity issue in multilingual context.

3.3.2.2 Linguistic engineering principles

MTT lexicon contains to our knowledge the richest lexical information. This knowledge is structured according to a very precise format extensively described in the literature [Mel'cuk,1984]. In addition to the lexical information contained in UCG lexicon, MTT lexicon expresses the inter definition of lexemes using two powerful means : the decomposition of lexeme meaning by other lexemes and the use of lexical functions which accounts for the idiomatic aspect of meaning structure in a given language. Lexical descriptions have been carried out for various languages and more extensively for French and Russian.

Extensive description of English syntax in MTT have been published in [Mel'cuk, 198X]. Parts of French syntax have also been studied for application purposes [FoG, GLOSE, Moose]. An evaluation of the current status of the theory has just been made taking into account the requirements issued from the corpus study for Hyperdocsy [cf 3.2.2]. At present, MTT is missing adequate devices for handling complex quantifications, plurals and negation. The reason is that the theory has until recently not paid much attention to representations of scope in semantic networks. Current work is however turned towards answering this need by exploring different approaches.

MTT provides methodological guide-lines for describing lexical knowledge and grammatical knowledge by a precise specification of knowledge levels. MTT lexicography method is illustrated by the construction of the Contemporary French Combinatorial Dictionary [See DECFC principles]. In general, the model gives precious indication on where to locate the adding of new knowledge for incrementing the linguistic description.

Several applications of real text generation (Gossip, FoG89) were developed. FoG produced bilingual weather reports. Other applications are currently in progress (generation of explanations).

Control of the different knowledge sources that must intervene in text generation is still a research issue. Applications with MTT have only provided solutions to this issue in narrow applications and the search for more general solutions remains open. However, examples from the documentation have already been analyzed in the MTT framework with the aim of specifying how they can be generated. A strong point for MTT is that the framework so far does not need important changes and extensions are performed by only adding new linguistic rules to the grammar. Also, semantic structuring of MTT potentially allows a paraphrasis functionality more complete than the traditional syntactic paraphrasis performed by existing systems.

3.3.2.3 Multilingual generation engineering principles

Multilingual generation is a complex issue which has not been very often dealt with in real applications. An obvious approach to generating the same text in different languages will put as many different generators in parallel as languages considered. If generating different languages does not question the theoretical framework, it is already an argument in favour of the multilingual capability of the model.

One may also wish to have a model where some parts of the generator are common throughout the different languages considered. Of course, the assumption is that such common treatments or levels of representation may be common to different languages. Some may deny this assumption. In fact, deepest levels of text representation, like the one produced by text planning can be common except maybe when considered languages show important cultural differences. This fact is strengthened by similarities observed between texts belonging to the same technical sub-language in different languages [Kittredge, 1982]. In this case, applications may involve the same conceptual and semantic structures (FoG89) for different languages like English and French.

MTT is particularly well suited for satisfying the requirements of multilingual generation stated above. Because of its stratificational property, the theory is able to isolate levels of analysis that are common to different languages from those that contain the specificity of the different languages. For some sub-languages, the SemR of MTT constitutes a good interlingua from which peculiar structures of different languages are derived. In general, the previous comparative studies of languages (Russian, English, French, Spanish) prevents the theory from having a framework too narrow to handle different languages. Devices such as lexical functions and paraphrasis rules have already been tested for several languages and demonstrated their trans-linguistic value.

3.4. Conclusion

On the one hand, evaluation of MTT and UCG according to corpus analysis results concluded that both models have the capacity for synthesizing sentences present in technical documents.

On the other hand, their evaluation using the more general criteria stated above has just started. Final conclusions will be provided in the very first weeks of a future project. Beforehand, operational procedures that rule out the applying of qualitative criteria to our linguistic models have to be defined.

We intend to end up this work with a first but adequate evaluation framework for generation applications. With the help of this result, we want to be able to evaluate as well other linguistic models which are sufficiently documented.

3.5. References

Mel'cuk I. et al., *Dictionnaire Explicatif et Combinatoire du Français Contemporain*, Vol. I et II, Presses de l'Université de Montréal, Montréal, 1984.

Mel'cuk I. & Pertsov N., *Surface Syntax of English*, Benjamins, London, 1987.

Kittredge R. & Lehrberger J., *Sublanguages. Studies of Languages in Restricted Semantic Domain*, 1982.

APPENDIX A

1 Introduction.

Overview(PMS)
1.1 Purpose and Scope of Power Management System.
Level1-functions(PMS)
The Power Management System (PMS) monitors and controls electricity production
Level1-structure(PMS)
via four Generating Sets (GS), that is three Diesel Generators (DG) and one Shaft Generator (SG).

Overview-main-components(PMS)
Level1-structure(SG)
The SG is connected to the Main Engine (ME)
Level1-functions(SG)
and it can produce power to either the busbar or the Bow-/Stern-Thruster (BT,ST).
Level1-structure(DG)
()
Level1-functions(DG)
()
Level2-functions(DG)
The DG part of the system is a standardized full-automatic start/stop, synchronization, frequency control, loadsharing and black out start system.
Level2-functions(SG)
The SG part include synchronization to busbar (BB) and automatic connection of SG to BT/ST.

Structural-description(PMS)
1.2 Overview of the controlled components/system.

Ref-to(figure1.a)
In figure 1.a is shown schematic the controlled/monitored system.
List-of-components(PMS)
This include the GS, the GS Main Breakers (MB), the BT/ST MB's and the Emergency Switch Board (ESB).
Level1-functions(PMS)---------> should be in 1.1
Furthermore the PMS monitors alarms from the alarm system, all alarms detected by the PMS system and information for the DG surrounding machinery.

Functional-description(PMS)
2 Functional description for normal use.

Overview(control)
2.1 Control modes in general.

Number(control-modes)
The PMS contains three modes of operation for DGs and three modes of operation for the SG - they are explained briefly below:
MANUAL:

Command-device(control-modes)

Each GS has a MANUAL/AUTO selector.

Effect(device)

When MANUAL mode is selected it overrides the two other modes AUTOMATIC and SEMIAUTOMATIC.

Description(control-modes)

Description(manual-mode)

Actions(manual-mode,DG,PMS)

DGs: *No control at all of DG in question.*

Actions(manual-mode,SG,PMS)

SG: *MB to BT,ST:* *No control at all of thruster MB in question.*

MB to Busbar (BB): *No control at all of MB in question.*

Description(auto-mode,DG)

The next two modes only concern operation of DGs. These modes are common modes for all DGs.

The modes require, that the DGs are in AUTO mode (not MANUAL).

Description(SA-mode,DG)

DG SEMIAUTOMATIC:

actions(SA-mode,DG,PMS)

The PMS will automatically perform the following functions:

1 *Black out start.*
2 *Loadsharing and frequency control of online DGs.*
3 *Only one start attempt in case of starting failure.*
4 *Synchronization, when the diesel engine is started.*

actions(SA-mode,DG,operator)

Start and stop of DGs, except during black out start, is commanded by the operator.

Description(A-mode,DG)

DG AUTOMATIC:

actions(A-mode,DG,PMS)

The PMS will automatically perform the functions 1 - 4 described for DG SEMIAUTOMATIC-mode, and the following functions:

1 *Start and stop of DGs based on actual power requirements.*

2 *Change to the next DG in the standby sequence, if a DG does not start.*

3 *Start of standby DG and shut down of faulty DG on AE prewarnings.*

4 *Start of one or two DGs (load dependent), when SG is wanted stopped either because mode is changed to a mode without SG on the ship handling mode selector (ref.[2]) (SG AUTOMATIC mode only) or by command from the ISC consoles (SG SEMIAUTOMATIC mode only).*

5 *Start of two DGs if SG online has a standby start shut down upon ME slowdown or if SG frequency is above/below allowed range for BB operation.*

actions(A-mode,DG,operator)

()

Description(Auto-mode,SG)

The next two modes only concern operation of SG.

Relationship-between(DG,SG,Auto-mode)

Operation of DG is independent of selected mode SG SEMIAUTOMATIC and SG AUTOMATIC.

Description(SA-mode,SG)

SG SEMIAUTOMATIC:

actions(SA-mode,SG,PMS)

The PMS will perform the following functions:

1 Synchronization of SG to BB.

2 Immediately stop of DGs online after SG MB to BB is closed.

3 Start sequence for switching BT/ST online.

4 Stop sequences for switching SG offline from either BB or thruster.

actions(SA-mode,SG,operator)

Start and stop of SG's to either BB or BT/ST is commanded by the operator.

Description(A-mode,SG)

SG AUTOMATIC:

actions(A-mode,SG,PMS)

The PMS will automatically perform the functions 1 - 4 described for SG SEMIAUTOMATIC-mode, and the following functions:

1 Automatic control of SG to either BB or BT/ST dependent of mode selected on ship handling mode selector (ref[2]).

actions(A-mode,SG,operator)

()

Operations(control)

2.2 PMS operation strategy.

Operations(control,DG)

Action-description(Blackout-start,DG)

Conditions(Blackout-start,_,_,DG,_)

Blackout start is enabled when at least one DG is in AUTO-mode and not blocked

Definition(blocked,DG)

(blocked means that the DG is not available f.x. because of an alarm).

level1-process(Blackout-start,DG)

One of two actions will take place after a blackout:

1 If one or more DGs are running the highest priorited will be switched online when its frequency has reached a preset level.

2 If no DG is running, the first in the standby sequence will be started and switched online, when its frequency has reached a preset level.

The next DG in the standby sequence will be started if the former DG fails to start or switch online.

Definition(switch-online,DG)

Switch online means in this case direct connection without synchronization of MB to BB commanded by the PMS system.

Action-description(priority-decision,DG)

Conditions(priority-decision,_,_,DG,_)

To decide the master/standby sequence of the DGs each DG always has a priority. This is either default or selected from the ISC consoles.

Priorities are:

1 Master

2 1. standby

3 2. standby

Level1-process(priority-decision,DG)

The priority sequence is used in the PMS control modes to:

DG SEMIAUTOMATIC: Select which online DG is frequency *controlled.*

DG AUTOMATIC: Select the DG which is supposed always to be online and the following standby sequence of DGs.
Furthermore to select which online DG is frequency controlled.

Summary(priority-decision,DG)
The priority is used to select which DG is started in case of black out.

Operations(control,SG)
()

Level1-description(control)
2.3 Diesel generator control.

Level1-description(control,DG)
agents(control,DG)
The DGs can be controlled direct on the Auxiliary Engine (AE), from the MSB or from the PMS.
command-device(control)
Switching between the different control possibilities is done with a switch, named MANUAL/AUTO, mounted in the MSB.
Effect(device)

Level1-description(control,manual-mode,DG,_)
agents(control,manual-mode,DG,_)
When the MANUAL/AUTO switch is in MANUAL position, the DG is controlled either from the MSB or directly on the AE.
Level1-actions(control,manual-mode,DG,_)
Synchronizing, closing/breaking of the MB and speed/load control is done from the MSB. This is called MANUAL mode.

Level1-description(control,auto-mode,DG,_)
agents(control,auto-mode,DG,_)
When the MANUAL/AUTO switch is in AUTO position, the DG is said to be under PMS control. In this situation the basic control is performed from the ISC-system.
Level1-actions(control,auto-mode,DG,_)
()

Level1-description(control,SA-mode,DG,_)
DG SEMIAUTOMATIC mode.
Agents(control,SA-mode,DG,_)
Functions(control,SA-mode,DG,operator,_)
The operator controls from the ISC consoles, which DGs are online and stopped. The operator controls in other words the available power.
Functions(control,SA-mode,DG,PMS,_)
The online, PMS controlled DG with highest priority is frequency controlled. This is called the master DG.
Alarm-rules(SA-mode,DG)
In case a critical condition, which could lead to a shut down, occurs, an alarm will be indicated.
Level1-actions(PMS, DG,SA-mode,_)
Loadsharing is performed between all online, PMS controlled DGs.
Level1-actions(operator,DG,SA-mode,_)
If the operator wants to stop an online, PMS controlled DG, this can be done from the ISC consoles (Note: This can not be done

with the master DG without changing its priority).

Definition(stopping)

Stopping means deloading, switching offline and stopping of engine.

In the same way start of a stopped DG can be done from the ISC console.

Definition(starting)

Starting means starting of engine, synchronization and switching online.

Level1-description(control,A-mode,DG,_)

DG AUTOMATIC mode.

Agents(control,A-mode,DG,_)

()

Functions(control,A-mode,DG,PMS,_)

The DG with the highest priority, under PMS control and not blocked is always online and master DG (if SG operation to BB is not selected).

The following DGs are started, synchronized and switched online, respectively deloaded, switched offline and stopped automatically all dependent of their priority and the actual power consumption.

Level1-actions(PMS, DG,A-mode,_)

Loadsharing of all online, PMS controlled DGs is also part of the AUTOMATIC mode.

If a PMS controlled DG is wanted out of the automatic start/stop sequence, this can be done by switching it to MANUAL mode.

If stop is wanted on an online, PMS controlled DG without changing its mode, it can be done by changing the priority, so that the online DG gets a lower priority. The PMS will then automatically update the plant, i.e. start a new DG with higher priority and then stop the one in question.

In the same way start of a stopped, PMS controlled DG can be done by changing the priority so that the stopped DG gets a high priority.

Operation-rules(control, DG, auto-mode)

If the PMS control mode is changed from SEMIAUTOMATIC to AUTOMATIC the plant will automatically update to the present priority sequence.

Alarm-rules(auto-mode,DG)

In case an alarm for standby start occur, a standby DG is started. Then the faulty DG is stopped and blocked.

Level1-description(control,SG)

2.4 Shaft generator control.

Overview(control,SG)

The SG can connect to either BB or BT/ST,
It is impossible to connect the SG to thruster(s) and to the BB at the same time.
When the SG is connected to BB, the BB frequency depends on the ME RPM.

Agents(control,SG)

and controls are performed either from the PMS-system or from the MSB.

Command-device(control)

Switching between control possibilities is performed with a switch named MANUAL/AUTO, mounted in the MSB.

effect(device)

Level1-description(control,manual-mode,SG,BB)
Agents(control,SG,manual-mode,BB)
level1-actions(control,SG,manual-mode,BB)

SG to BB: When the MANUAL/AUTO switch is in MANUAL position, synchronization, closing/breaking of the MB is done from the MSB. This is called MANUAL mode.

Process(synchronization,manual-mode,operator,DG,BB)

Note: In order to synchronize the DGs must be switched to MANUAL and synchronization is done by adjusting BB frequency with online DGs. When the SG MB is closed, the operator must stop the DGs manually.

Level1-description(control,auto-mode,SG,BB)

Agents(control,SG,auto-mode,BB)

When the MANUAL/AUTO switch is in AUTO position, the SG MB is said to be under PMS control.

level1-actions(control,SG,auto-mode,BB)

Synchronization is performed by the online DGs automatically.

Process(synchronization,auto-mode,PMS,SG,BB)

Note: this requires that the DGs online is under PMS control too. After the SG MB is closed, the DGs are stopped automatically.

Level1-description(control,manual-mode,SG,Thrusters)

Agents(control,manual-mode,SG,Thrusters)

SG to BT/ST: When the AUTO/MANUAL switch is in MANUAL position, the MB to the BT/ST is controlled from the MSB.

level1-actions(control,manual-mode,SG,Thrusters)

This includes control of SG voltage during power up of BT/ST.

Level1-description(control,auto-mode,SG,Thrusters)

Agents(control,auto-mode,SG,Thrusters)

level1-actions(control,auto-mode,SG,Thrusters)

When the AUTO/MANUAL switch is in AUTO position, the MB and the power up procedure for ST/BT are controlled by the PMS. In this situation the BT/ST MB is said to be under PMS control.

Alarm-rule(auto-mode,Thrusters)

Upon ME slowdown or missing thruster hydraulic pressure, the PMS will open the thruster(s) MB(s).

Level2-description(control,SA-mode,SG,BB)

SG SEMIAUTOMATIC mode.

Shaft generator connected to Busbar.

Agents(control,SA-mode,SG,BB)

Functions(control,SA-mode,SG,operator,BB)

In this mode, the operator can connect/disconnect SG to/from the BB via the ISC console.

Process(connection,SA-mode,operator,SG,BB)

Connection means:

1. *Frequency controlled DG synchronize BB to SG.*
2. *SG is connected to BB.*
3. *DGs deloads.*
4. *DGs are disconnected and stops.*

Process(disconnection,SA-mode,operator,SG,BB)

Disconnection means:

1. *One or two DG(s) start and switch online (this is only performed automatically, if the DGs are in DG AUTOMATIC mode. In DG SEMIAUTOMATIC it is the operator's responsibility to start the DGs after the SG is commanded to stop).*
2. *DG takes load i.e. SG deloads.*
3. *SG is disconnected.*

Conditions(connection,SA-mode,operator,SG,BB)

In order to connect SG to the BB, the following conditions must be satisfied:

1. *The frequency of the SG is in a range near to normal BB frequency.*
2. *The ME is locked to fixed RPM.*
3. *DGs are connected to the BB.*

4 *SG is not connected to BT or ST.*

If one or more of these conditions is not satisfied, the PMS will not connect the SG to the BB.

Level2-description(control,SA-mode,SG,Thrusters)
Shaft Generator connected to thrusters.
Agents(control,SA-mode,SG,Thrusters)
Functions(control,SA-mode,SG,operator,Thrusters)
In SG SEMIAUTOMATIC mode, the operator can connect and disconnect the SG's to/from its BT and/or ST.
Process(connection,SA-mode,operator,SG,Thrusters)
Connection to BT and/or ST means:
1 *SG is deexited.*
2 *SG is switched to current mode.*
3 *SG MB to thruster(s) is closed. SG is exited.*
4 *When thruster is running SG is switched to voltage mode.*
5 *Thruster(s) is ready for operation, when thruster current is at idle level.*
Conditions(connection,SA-mode,operator,SG,Thrusters)
Before connection of BT and/or ST, the following conditions must be satisfied:
1 *The frequency of the SG is in the correct range for operation of the BT or ST.*
2 *The speed of the ME is limited to the speed operating range for BT/ST.*
3 *The SG is not connected to BB.*
4 *The BT/ST has pitch on zero.*
5 *The BT/ST has correct hydraulic pressure.*

If one or more of these conditions is not satisfied, the PMS will not connect the SG to the thruster.
Process(disconnection,SA-mode,operator,SG,Thrusters)
Disconnection of BT and/or ST means SG MB to BT respectively ST is opened.
Conditions(disconnection,SA-mode,operator,SG,Thrusters)
Before disconnection of the BT/ST, the pitch must be in zero position. No disconnection is performed before this is satisfied.

Level2-description(control,A-mode,SG,_)
SG AUTOMATIC mode.
Agents(control,A-mode,SG,_)
In this mode, the SG operation is fully automatic and controlled from the ship handling mode selector (ref.[2]).
Functions(control,A-mode,SG,operator,_)
It is impossible to connect/disconnect the SG via the ISC console neither to the BB nor to the thruster(s).
Process(connection,A-mode,SG,PMS,_)
Process(disconnection,A-mode,SG,PMS,_)
The sequences for connection and disconnection to BB and to thruster(s) is described in previous section.
Conditions(connection,A-mode,SG,PMS,_)
If one of the conditions (see previous section) is not satisfied, the PMS system will ignore a request for connection.

Overview(power-reservation)
2.5 *Power reservation.*
Agents(Power-reservation,consumer)
The PMS system controls power reservation for heavy consumers.
Process(Power-reservation,_,PMS,consumer,_)
The power reservation operates with a start request/running input from the power consumer and a start blocking output to the power consumer.
The maximum power consumption for the consumer is known by the PMS system.
Level1-process(power-reservation,consumer)

The start blocking is activated while the req./running signal is inactive.

Upon request, when the available power on the BB is below the consumer maximum power consumption, the start blocking continue active until a standby DG is started and switched online. Then the start blocking is released to the consumer. When started, the power consumer must maintain the start request to the PMS - the PMS system treats it now as a consumer running signal.

When the consumer stops, the start req./running signal is removed.

Because of great variation in power consumption, the PMS system measures the actual power consumption of each consumer.

When the consumer is running, the PMS system reserves the maximum power for the consumer minus the actual consumer power consumption. This means, that the available power on the BB always will be greater than full load on the consumer.

Appendix B

APPENDIX B1: List of predicates and arguments

accept
    accept(_,command)
activate
    activate(GS,action)
    activate(_,signal/process/action)
    activate(_,blocking,while-state)
adjust
    adjust(_,frequency,with-GS)
affect
    affect(blocking,calculation)
allocate
    (_,GS-block,in-unit)
allow
    allow(_,range,for-operation)
assume
base
    base(_,action,on-requirement)
be
    be-online(GS)
    be-part-of(action,mode)
    be-in-position(pitch,value)
block
    block(_,GS,for-time)
call
change
    change(_,alpha-unit)
    change(_,mode)
    change(_,mode,on-selector)
    change(_,mode,from-mode,to-mode)
close
    close(_,component(MB),to-component)
come from
    come-from(command,component)
command
    command(operator,actions/functions)
    command(_,GS,to-action)
concern
    concern(mode,operation-of-GS)
connect
    connect(GS,to-component)
    connect(PMS,GS,to-component)
    connect(operator,GS,to-component)
    connect(_,power,to-component)
    connect(_,GS,to-component/engine)
    connect(operator,GS,to-component,via-component)
    connect(_,GS,to-component,via-component/device)
consist
    consist(PMS,of-blocks)
    consist(block,of-module)
contain
    contain(GS-alpha,system)
    contain(PMS,modes-of-operation)
continue
    continue(GS-alpha,operation)

control
    control(operator,power)
    control(PMS,power-reservation)
    control(module,module)
    control(module,status)
    control(operator,state,from-component)
    control(_,GS,from/on-component)
    control(_,GS-operation,from-device/component)
    control(_,module,from-module)
deexit
    deexit(_,GS)
deload
    deload(GS)
    deload(module,MB)
    deload(_,DG,to-load)
demand
    demand(_,GS,offline)
depend
    depend-on(frequency,ME RPM)
describe
    describe(_,block)
    describe(_,condition/exception)
    describe(_,function,for-mode)
    describe(_,sequence-of-actions,in-text)
detect
    detect(PMS,alarm)
    detect(_,alarm)
disable
disconnect
    disconnect(_,GS)
    disconnect(operator,GS,to-component)
    disconnect(operator,GS,to-component,via-component)
    disconnect(_,GS,to-component,via-component/device)
do
    do(_,action,with-device/component)
    do(_,action,from-component)
    do(_,action/state,by-action)
    do(_,action,via-module)
download
    download(_,alpha-unit,with-code)
drop
    drop(power,below-limit,for-time)
enable
    enable(_,function)
    enable(module,module,for-action)
    enable(_,module,via-module)
    enable(_,synchronizer,for-action)
    enable(reset,alpha-unit,for-operation)
exceed
    exceed(power,limit,for-time)
exit
    exit(_,GS)
explain
fail
    fail(action/command)
    fail(GS,to-action)
feed
    feed(_,alarm/information,through-module)
follow
    follow(numbering,layout)

gate
    gate(_,frequency,from-module,to-module)
    gate(_,error,from-module,to-GS-alpha,via-module)
generate
    generate(system,warning)
    generate(_,failure)
    generate(_,alarm)
    generate(module,signal,to-GS)
    generate(module,request,to-GS)
    generate(module,alarm,to-module)
    generate(failure,failure,to-system)
    generate(_,alarm,to-system)
    generate(_,command,in-module)
    generate(_,failure,to-module)
    generate(_,clock,to-PMS-alpha,for-time)
get
    get(GS,priority)
give
handle
    handle(module,input/information)
happen
have
    have(GS,selector)
    have(GS,priority)
    have(GS,shutdown)
    have(component,pitch/pressure)
ignore
    ignore(PMS,request)
include
    include(part-of-system,action/function)
    include(PMS,set-of-components)
    include(failure,blocking)
    include(data-flow,data-flow/interface)
indicate
    indicate(_,alarm)
initialize
    initialize(_,PMS)
keep track
know
    know(PMS,power-consumption)
latch
    latch(_,alarm)
lead
    lead(condition,state/action)
limit
    limit(_,speed,to-speed-range)
list
lock
    lock(_,engine,RPM)
maintain
mean
measure
    measure(PMS,power-consumption)
mention
    mention(_,nothing,about-GS-status)
monitor
    monitor(PMS,action,via-GS)
    monitor(PMS,information,from-system/machinery)
mount
    mount(_,switch,in-component)

occur
    occur(condition)
    occur(alarm)
    occur(action)
open
    open(PMS,MB)
    open(module,MB)
operate
    operate(power-reservation,with-command-input) ?
override
    override(mode,mode)
perform
    perform(PMS,functions)
    perform(_,action)
    perform(module,action)
    perform(_,action,from-component)
    perform(_,action,by-component)
    perform(_,action,with-device/component)
    perform(_,calculation,in-module)
prepare
    prepare(_,module,for-action)
process
    process(module,message)
produce
    produce(GS,power,to-component)
raise
    raise(_,frequency)
reach
    reach(frequency,value/level)
    reach(GS,frequency)
receive
    receive(_,command)
    receive(_,failure,from-GS)
    receive(module,signal/data,from-GS)
release
    release(_,process,to-consumer) ?
    release(GS,command,to-GS)
remove
    remove(_,signal)
require
    require(mode,that-state)
    require(action,that-state)
reservate
    reservate(PMS,power,for-consumer)
reset
    reset(_,GS,from-component)
restart
result
    result(command,in-action)
run
    run(consumer)
    run(GS)
satisfy
    satisfy(_,condition)
say
scale
    scale(_,input,by-connos)
secure
    secure(module,GS-state)
select

select(_,mode)
select(_,mode,on-selector)
select(_,GS,in-module)
select(_,GS,with-priority)

send
send(_,failure,to-module)
send(_,request,to-GS-alpha)

show
show(_,schema,in-figure)

start
start(GS)
start(PMS,GS)
start(_,GS)
start(_,GS,from-component,in-mode)

stop
stop(consumer)
stop(PMS,GS)
stop(operator,GS)
stop(system,engine)
stop(module,GS,via-module)
stop(_,GS,from-component)

succeed
succeed(action)
succeed(action,within-time)

supervise
supervise(module,status)

surround
surround(machinery,GS)

switch
switch(_,GS)
switch(_,between-possibilities)
switch(_,GS,to-mode)

synchronize
synchronize(GS,component,to-GS)
synchronize(_,GS,to-component,via-module)

take
take(GS,load)

take-into-account
take-into-account(module,priority/status)

take place
take-place(action)
take-place(action,after-action)

transfer
transfer(_,value,PMS-alpha)

transmit
transmit(module,information/data,to-GS(-alpha))

treat
treat(PMS,consumer,signal) ?

update
update(PMS,plant)
update(plant,to-priority)

use
use(_,module)
use(_,priority,to-action)
use(_,frequency,synchronisation)
use(_,priority,to-action)

vary
vary(_,frequency)

want
want(operator,to-action)

```
want(GS,to-action)
want(_,action,on-GS)
want(_,GS,out-of-sequence)
```

APPENDIX B2: DETERMINERS AND VALUES

Values

N	new item	(cf Halliday "exophoric")
A	anaphoric	("endophoric")
K	cataphoric	("endophoric")
U	unicity	
T	totality	
C	collective	
D	distributive	
NUM	number	

1.1	0 purpose		
	0 scope		
	0 PMS		
1.1.1	the PMS	A	1.1
	0 electricity production		
	four GSs	N,NUM	
	three DGs	N,NUM	
	one SG	N,NUM	
1.1.2	the SG	A	1.1.1
	the Main Engine	N,U	
	it	A	SG
	the busbar	N,U	
	the Bow-/Stern-Thrusters	N	
1.1.3	the DG part	A ; N,C	
	the system	A	PMS
	a system	REPETITION	
1.1.4	the SG part	A ; N,C	
	0 synchronization		
	0 busbar		
	0 connection		
	0 SG		
	0 BT/ST		
1.2	0 overview		
	the components/system	K	1.2.2
1.2.1	0 figure		
	the system	K	1.2.2
		A	TITLE 1.2
1.2.2	this	A	the system
	the GS	A	
	the GS MBs	N,D	
	the BT/ST MBs	N,D	
	the ESB	N,U	
1.2.3	the PMS	A	
	0 alarms		
	the alarm system	N,U	
	all alarms	N ,T	all the
	the PMS	A	

	0 information		
	the DG surrounding machinery	N	the machinery which...
2	0 description		
	0 use		
2.1	0 modes		
2.1.1	the PMS	A	
	three modes	N,NUM	
	0 operation		
	0 DGs		
	three modes	N,NUM ou A ?	
	AMBIGUITY		
	0 operation		
	the SG	A	
	they	A	three or six modes ?
2.1.2	each GS	D	
	a M/A selector	one/its/its own	
2.1.3	0 M-mode		
	it	A	M-mode
	the two other modes	A	three modes 2.1.1
2.1.4	0 DGs		
	no control		
	0 DG in question	S	
2.1.5	0 SG		
	0 MB		
	0 BT,ST		
	no control		
	0 thruster MB		
2.1.6	0 MB		
	0 BB		
	no control		
	0 MB		
2.1.7	the next two modes	A	three modes 2.1.1
		K	
	0 operation		
	0 DGs		
2.1.8	these modes	A	2.1.7
	0 modes		
	all DGs	T;D	are the same for each
2.1.9	the modes	A	2.1.7, 2.1.8
	the DGs	T	all DGs
	0 A-mode		
	0 M		
2.1.10	the PMS	A	
	the following functions	K	2.1.11-14
2.1.11	0 blackout start		

2.1.12	0 loadsharing		
	0 frequency control		
	0 DGs		
2.1.13	only one start attempt	NUM	
	0 starting failure		
2.1.14	0 synchronization		
	the diesel engine	D	DGs
2.1.15	0 start		
	0 stop		
	0 DGs		
	0 blackout start		
	the operator	N,U	
2.1.16	the PMS	A	
	the functions	A	2.1.10
	0 DG		
	0 S-mode		
	the following functions	K	2.1.17-21
2.1.17	0 start		
	0 stop		
	0 DGs		
	0 power requirements		
2.1.18	0 change		
	the next DG	S	
	the standby sequence	N,U	
	a DG	non-S	any
2.1.19	0 start		
	0 standby DG		
	0 shutdown		
	0 faulty DG		
	0 AE prewarnings		
2.1.20	0 start		
	one or two DGs	NUM, non-S, PART	
	0 load		
	0 SG		
	0 mode		
	a mode		
	0 SG		
	the ship handling mode selector	N,U	
	0 SG A-mode		
	0 command		
	the ISC consoles	N	
	0 SG S-mode		
2.1.21	0 start		
	two DGs	NUM, PART	
	0 SG		
	a standby start		
	0 ME slowdown		
	0 SG frequency		
	0 range		
	0 BB operation		

2.1.22	the next two modes	K,PART	2.1.24-31
	0 operation		
	0 SG		
2.1.23	0 operation		
	0 DG		
	0 selected mode		
2.1.24	the PMS	A	
	the following functions	K	2.1.25-28
2.1.25	0 synchronization		
	0 SG		
	0 BB		
2.1.26	0 stop		
	0 DGs		
	0 SG MB		
	0 BB		
2.1.27	0 start sequence		
	0 switching		
	0 BT/ST		
2.1.28	0 stop sequence		
	0 switching		
	0 SG		
	0 BB		
	0 thruster		
2.1.29	0 start		
	0 stop		
	0 SG		
	0 BB		
	0 BT/ST		
	the operator		
2.1.30	the PMS	A	
	the functions	A	2.1.24-28
	0 SG mode		
	the following functions	K	2.1.31
2.1.31	0 control		
	0 SG		
	0 BB		
	0 BT/ST		
	0 mode		
	0 ship handling mode selector		
2.2	0 PMS operation strategy		
2.2.1	0 blackout start		
	at least one DG	NUM,PART	
	0 A-mode		
	the DG	A	
	an alarm	non-S	
2.2.2	one of two actions	K,PART	

	a blackout	non-S		
2.2.3	one or more DGs	PART		
	the highest prioritied	S	DG	
	its frequency	A,POSS		the highest
	a preset level	non-S		
2.2.4	no DG			
	the first	S	DG	
	the standby sequence	A	2.1.18	
	its frequency	A	the first	
	a preset level	non-S	no reference with 3	
2.2.5	the next DG	S		
	the standby sequence	A		
	the former DG	S	next/former	
2.2.6	0 switch online			
	this case	A	2.2.2 blackout	
	0 connection			
	0 synchronization			
	0 MB			
	0 BB			
	the PMS	A		
2.2.7	the master/standby sequence	N ; A ?		2.1.18
	the DGs	T		
	each DG	D		
	a priority	N,D	each DG	
2.2.8	this	A	2.2.7 decide...	
	the ISC consoles	A	2.1.20	
2.2.9	0 priorities			
	0 master			
	0 standby 1			
	0 standby 2			
2.2.10	the priority sequence	A		2.2.7, 2.1.18
	the PMS control modes	A	2.1	
2.2.11	which DG	A,S	bad	
2.2.12	the DG	A,S		
	the following standby sequence	A	bad	
	0 DGs			
2.2.13	which DG	A,S		
2.2.14	the priority		bad	
	which DG		bad	
	0 blackout			
2.3	0 diesel generator control			
2.3.1	the DGs	T		
	the AE	N,U		
	the MSB	N,U		
	the PMS	A		

2.3.2	0 switching			
	the different control possibilities	A		2.1, 2.2.10
	a switch	N,U,non-S		
	the MSB	A		2.3.1
2.3.3	the M/A switch	A		2.3.2
	0 M-position			
	the DG	D;T		all DGs
	the MSB	A		
	the AE	A		2.3.1
2.3.4	0 synchronizing			
	0 closing			
	0 breaking			
	the MB	D		DGs 2.3.3
	0 speed/load control			
	the MSB	A		
2.3.5	this	A	2.3.3	"when..."
	0 M-mode			
2.3.6	the M/A switch	A		
	0 A-position			
	the DG	D;T		all DGs
	0 PMS control			
2.3.7	this situation	A		2.3.6
	the control	N;A		2.3.6?
	the ISC system	N;A		ISC system?
2.3.8	the operator	A		
	the ISC consoles	A		
	which DGs	S		bad
2.3.9	the operator	A		
	other words	DIFF		
	the power	N		
2.3.10	the online DG	S,U		
	0 highest priority			
2.3.11	this		bad : this one/this DG	
	the master DG	U		
2.3.12	a condition	non-S		
	a shutdown	non-S		
	an alarm	non-S		
2.3.13	0 loadsharing			
	all DGs	T		
2.3.14	the operator	A		
	a DG	non-S		
	this	A		"want to ..."
	the ISC consoles	A		
2.3.15	this	A		2.3.14
	the master DG	S		
	its priority	POSS		DG

2.3.16 0 stopping			
0 deloading			
0 switching			
0 stopping			
0 engine			
2.3.17 the same way	COMPAR		2.3.14
0 start			
a DG	non-S		
the ISC consoles	A		
2.3.18 0 starting			
0 starting			
0 engine			
0 synchronization			
0 switching			
2.3.19 the DG	S,U		
the highest priority	S		
0 PMS control			
0 master DG			
0 SG operation			
0 BB			
2.3.20the following DGs	S		
all	A,T		DGs
their priority	D		DGs
the power consumption	N,S		
2.3.21 0 loadsharing			
all DGs	T		
the A-mode	A		
2.3.22 a DG	non-S		
the start/stop sequence	A		
this	A		"DG wanted..."
it	A		DG
0 M-mode			
2.3.23 0 stop			
a DG	non-S		
its mode	POSS		DG
it	A		"if stop..."
the priority	G		
the DG	S		
a priority	POSS		DG
2.3.24 the PMS	A		
the plant	N		
a new DG	non-S		
0 higher priority			
the one in question	A		2.3.23 DG
2.3.25 the same way	COMPAR		2.3.23-24
0 start			
a stopped DG	non-S		
the priority	A	its	DG
the DG	A		a stopped DG
a priority	A		DG

2.3.26	the PMS control mode	A			
	0 SEMI				
	0 AUTO				
	the plant	A			
	the present priority sequence	S			
2.3.27	an alarm	non-S			
	0 standby start				
	a standby DG	non-S			
2.3.28	the faulty DG	S	2.3.27		alarm
2.4	0 SG control				
2.4.1	the SG	A			
	0 BB				
	0 BT/ST				
	0 controls				
	the PMS	A			
	the MSB	A			
2.4.2	it	IMPS			
	the SG	A			
	0 thrusters		the		
	the BB	A			
	the same time	TEMP			
2.4.3	the SG	A			
	0 BB				
	the BB frequency	POSS		BB	
	the ME RPM	N			
2.4.4	0 switching				
	0 control possibilites				
	a switch	N			
	the MSB	A			
2.4.5	0 SG				
	0 BB				
	the M/A switch	A		2.4.4	
	0 M-position				
	0 synchronization				
	0 closing				
	0 breaking				
	the MB	S,A		SG to BB	
	the MSB	A			
2.4.6	this	A		2.4.5	
	0 M-mode				
2.4.7	the DGs	T			
	0 M				
	0 synchronization				
	0 BB frequency				
	0 DGs				
2.4.8	the SG MB	S			

	the operator	A	
	the DGs	T	
2.4.9	the M/A switch	A	
	0 A-position		
	the SG MB	T;D	MBs
	0 PMS control		
2.4.10	0 synchronization		
	the online DGs	S	
2.4.11	this	A	2.4.10
	the DGs online	A	2.4.10
	0 PMS control		
2.4.12	the SG MB	A	2.4.9
	the DGs	A	2.4.10
2.4.13	0 SG		
	0 BT/ST		
	the A/M switch	A	
	0 M-position		
	the MB	A	MBs
	the BT/ST	A	
	the MSB	A	
2.4.14	his	A	2.4.13
	0 control		
	0 SG voltage		
	0 power up		
	0 BT/ST		
2.4.15	the A/M switch	A	
	0 A-position		
	the MB	A	
	the power up procedure	A	2.4.14
	0 BT/ST		
	the PMS	A	
2.4.16	this situation	A	2.4.15
	the BT/ST MB	A	2.4.15
	0 PMS control		
2.4.17	0 ME slowdown		
	0 thruster hydraulic pressure		
	the PMS	A	
	the thrusters MBs	A	
2.4.18	0 SG		
	0 busbar		
	this mode	A	TITLE
	the operator	A	
	0 SG		
	the BB	A	
	the ISC consoles	A	
2.4.19	0 connection		
2.4.20	0 frequency controlled DG		

	0 BB			
	0 SG			
2.4.21	0 SG			
	0 BB			
2.4.22	0 DG			
2.4.23	0 DGs			
2.4.24	0 disconnection			
	one or two DGs	NUM		
	this	A	"start..."	
	the DGs	T;A	one or two DGs ?	
	0 A-mode			
2.4.25	0 S			
	it	IMPS		
	the operator's responsibility	N		
	the DGs	T;A	one or two DGs	
	the SG	A		
2.4.26	0 DG			
	0 load			
	0 SG			
2.4.27	0 SG			
2.4.28	0 SG			
	the BB	A		
	the following conditions	K		
2.4.29	the frequency	U		
	the SG	A		
	a range	non-S		
	0 BB frequency			
2.4.30	the ME	A		
	0 RPM			
2.4.31	0 DGs			
	the BB	A		
2.4.32	0 SG			
	0 BT			
	0 ST			
2.4.33	one or more of these conditions	A,NUM		2.4.29-32
	the PMS	A		
	the SG	A		
	the BB	A		
2.4.34	0 SG			
	0 thrusters			
	0 S-mode			
	the operator	A		

	the SG	A	
	its BT/ST	POSS	SG
2.4.35	0 connection		
	0 BT/ST		
2.4.36	0 SG		
2.4.37	0 SG		
	0 mode		
2.4.38	0 SG MB		
	0 thruster(s)		
2.4.39	0 SG		
2.4.40	0 thruster		
	0 SG		
	0 voltage mode		
2.4.41	0 thruster(s)		
	0 operation		
	0 thruster current		
	0 level		
2.4.42	0 connection		
	0 BT/ST		
	the following conditions	K	2.4.43-47
2.4.43	the frequency	S	
	the SG	A	
	the range	S	
	0 operation		
	the BT or ST	A	
2.4.44	the speed	S	
	the ME	A	
	the speed range	S	
	0 BT/ST		
2.4.45	the SG	A	
	0 BB		
2.4.46	the BT/ST	A	
	0 pitch		
	0 zero		
2.4.47	the BT/ST	A	
	0 hydraulic pressure		
2.4.48	one or more of these conditions	A,PART	2.4.43.47
	the PMS	A	
	the SG	A	
	the thruster	A	
2.4.49	0 disconnection		
	0 BT/ST		
	0 SG MB		
	0 BT		
	0 ST		

2.4.50	0 disconnection		
	the BT/ST	A	
	the pitch	A,POSS	their BT/ST
2.4.46			
	0 zero position		
2.4.51	0 disconnection		
	this	A condition	2.4.50 must
2.4.52	0 A-mode		
	this mode	A	TITLE
	the SG operation	S	
	the ship handling mode selector	A	
2.4.53	it	IMPS	
	the SG	A	
	the ISC consoles	A	
	the BB	A	
	the thruster(s)	A	
2.4.54	the sequences A		
	0 connection		
	0 disconnection		
	0 BB		
	0 thrusters		
	0 previous section	A	2.4.34-51
2.4.55	one of the conditions	A, PART	
	0 previous section	A	2.4.34-51
	the PMS	A	
	a request	non-S	
	0 connection		
2.5	0 power reservation		
2.5.1	the PMS	A	
	0 power reservation		
	0 consumers		
2.5.2	the power reservation	A	TITLE 2.5.1-2
	a start request	non-S	
	a running input	non-S	
	the power consumer	A D;T?	2.5.1 consumer
	a start blocking output	non-S	
	the power consumer	A	2.5.2
2.5.3	the power consumption	S	
	the consumer	A	2.5.2
	the PMS	A	
2.5.4	the start blocking	A	2.5.2 (output)
	the signal	A	2.5.2 start request running input
2.5.5	0 request		
	the power	S	

	the BB	A	
	the power consumption	S;D	
	the start blocking	A	2.5.4
	a standby DG	non-S	
2.5.6	the start blocking	A	2.5.5
	the consumer	A;D	
2.5.7	the power consumer	A	
	the start request	A	
	the PMS	A	
	the PMS	A	
	it	A	"start blocking"
	a consumer running signal	non-S	HYPERONYM/input
2.5.8	the consumer	A	
	the signal	A	
2.5.9	0 variation		
	0 power consumption		
	the PMS	A	
	the power consumption	A	
	each consumer	D	
2.5.10	the consumer	A;T	
	the PMS	A	
	the power	S	
	the consumer	A	
	the power consumption	A	2.5.9
2.5.11	this	A	2.5.10
	the power	A	2.5.10
	the BB	A	
	0 load		
	the consumer	A	2.5.10

APPENDIX B3 Syntactic analysis

Numbers refer to sentences in the text.

SYNTAX

1 - Active and passive

PREDICATES	ACTIVE	PASSIVE
accept		1
activate	3	1
adjust	1	
affect	1	
allocate		1
assume		1
block		13
call		3
change		3
close		7
come from	2	
command		3
concern	2	
connect	6	8
consist	3	
contain	3	
continue	3	
control	9	11
deexit		2
deload	3	2
demand		1
depend	2	
describe		3
detect		2
disable		1
disconnect	2	2
do		12
download		1
drop	2	
enable	2	5
exceed	1	
exit		2
explain		1
fail	6	
feed		4
follow	2	
gate		2
generate	14	25
get	2	
give	1	
handle	2	
happen	1	
have	1	
ignore	1	
include	7	
indicate		1
keep track	1	
know		1

latch		2
lead	1	
limit		1
list		1
lock		1
maintain	1	
mean	5	
measure	1	
mention		1
monitor	2	
occur	4	
open	3	2
operate	1	
override	1	
perform	5	14
prepare		1
process	1	
produce	1	
raise		1
reach	3	
receive		2
release	1	1
remove		4
require	2	
reservate	1	
restart		3
result	1	
run	2	
satisfy		6
say		1
scale		2
secure	1	
select		3
send	1	5
show		1
start	5	10
stop	5	14
succeed	3	
supervise	1	
switch		13
synchronize	2	3
take	1	
take place	1	
transfer		1
transmit	2	
treat	1	
update	2	
use		5
vary	1	1
want	2	6

Only active

adjust	1
affect	1
come from	2
concern	2
consist	3

contain	3
continue	3
depend	2
drop	2
exceed	1
fail	6
follow	2
get	2
give	1
handle	2
happen	1
have	1
ignore	1
include	7
keep track	1
lead	1
maintain	1
mean	5
measure	1
monitor	2
occur	4
operate	1
override	1
process	1
produce	1
reach	3
require	2
reservate	1
result	1
run	2
secure	1
succeed	3
supervise	1
take	1
take place	1
transmit	2
treat	1
update	2

Only passive

accept	1
allocate	1
assume	1
block	13
call	3
change	3
close	7
command	3
deexit	2
demand	1
describe	3
detect	2
disable	1
do	12
download	1
exit	2
explain	1
feed	4

gate	2
indicate	1
know	1
latch	2
limit	1
list	1
lock	1
mention	1
prepare	1
raise	1
receive	2
remove	4
restart	3
satisfy	6
say	1
scale	2
select	3
show	1
switch	13
transfer	1
use	5

Both active and passive

PREDICATES	ACTIVE	PASSIVE
activate	3	1
connect	6	8
control	9	11
deload	3	2
disconnect	2	2
enable	2	5
generate	14	25
open	3	2
perform	5	14
release	1	1
send	1	5
start	5	10
stop	5	14
synchronize	2	3
vary	1	1
want	2	6

- Passives

All the passives are without agent except

- OPERATOR	2.1.15, 2.1.29
- GS	2.4.10
- PMS	2.2.6, 2.4.15, 2.5.3

- Active vs passive forms

- PASSIVE	225
- ACTIVE	147

- Active verbs 47
- Passive verbs 55
- Active and passive verbs 21

TYPE OF VERB	VERB NUMBER	OCCURRENCES
Only active	43	83
Only passive	39	114
Both	16	174

VOICE	TOTAL
Active	147
Passive	224

2 - Cleft sentences

2.4.26

3 - Inversion subject-verb

1.2.1

4 - Impersonal form

"It is impossible to" 2.4.2, 2.4.53

5 - Relatives

- V + REL 2.2.11, 2.2.13,
2.2.14, 2.3.8
- N + REL 2.2.12, 2.3.12

6 - Completives

- *"require THAT"* 2.1.9, 2.4.11
- *"mean THAT"* 2.2.1, 2.5.11

7 - Infinitives

- infinitive + main
  - *TO* 2.2.7
  - *IN ORDER TO* 2.4.7, 2.4.28

- main + infinitive
  - *TO* 2.2.10, 2.2.14

8 - Prepositional phrases

POSITION	PREPOSITION	SENTENCE
Initial	*upon*	2.5.5
	because of	2.5.9
	before	2.4.50

Median	*in case of*	2.1.13
	during	2.1.15
	upon	2.1.21
	above	2.1.21
	below	2.1.21, 2.5.5 (-> F)
Final	*in case of*	2.2.14
	because of	2.2.1
	after	2.2.2
	during	2.4.14

PREPOSITION	POSITION
above	M
after	F
because of	I, F
before	I
below	M, F ?
during	M, F
in case of	M, F
upon	I, M

9 - Adverbials

We consider phrases such as 'in this situation' adverbials rather than prepositional phrases. They are always introduced by the preposition *'in'*.

POSITION	ADVERB	SENTENCE
Initial	*in figure*	1.2.1
	furthermore	1.2.3, 2.2.13
	in this situation	2.3.7, 2.4.16
	in the same way	2.3.17, 2.3.25
	then	2.3.28, 2.5.7
	in this mode	2.4.18, 2.4.52
Median	*that is*	1.1.1
	briefly	2.1.1
	only	2.1.7, 2.1.22,
2.4.24		
	automatically	2.1.10, 2.1.16,
2.3.20,		
		2.3.24, 2.3.26,
2.4.24 (F?)		
	in this case	2.2.6
	in other words	2.3.9
	respectively	2.3.20
	then	2.3.24
	fully	2.4.52
	now	2.5.7
Final	*in question*	2.1.4, 2.1.5, 2.1.6
	below	2.1.1
	at the same time	2.4.2
	manually	2.4.8
	automatically	2.4.11, 2.4.12

in previous section 2.4.54

ADVERB	POSITION
automatically	M, F
then	I, M

10 - Nominalizations

attempt
    attempt(action)
blocking
    blocking(action)
breaking
    breaking(MB)
calculation
    calculation(priority-sequence)
    calculation(power)
change
    change(GS,in-sequence)
closing
    closing(MB)
command
    command(from-component)
    command(action)
connection
    connection(GS,to-component)
    connection(MB,to-component)
    connection(_,to-component)
consumption
    consumption(power)
control
    control(GS)
    control(component)
    control(frequency)
    control(speed/load)
    control(GS,to-component)
deloading
    deloading(engine)
description
    description(module/interface/dataflow)
disconnection
    disconnection(_,of/to-component)
flow
    flow(data,in-system)
indication
    indication(description)
input
    input(_,from-component)
interface
    interface(to-system)
loadsharing
    loadsharing(GS)
    loadsharing(between-GS)
operation
    operation(GS)
    operation(component)

- order
  - order(action)
- output
  - output(_,to-component)
- production
  - production(electricity)
- request
  - request(for-action)
- requirement
  - requirement(power)
- reservation
  - reservation(power,for-consumers)
- selection
  - selection(priority,in-system)
- shutdown
  - shutdown(GS)
- slowdown
  - slowdown(engine)
- start
  - start(GS)
  - start(GS,to-component)
  - starting(engine)
- stop
  - stop(GS)
  - stop(GS,to-component)
  - stopping(engine)
- switching
  - switching(engine)
  - switching(between-possibilities)
  - switching(GS,from-component)
- synchronization
  - synchronization(component)
  - synchronization(frequency)
  - synchronization(GS,to-component)
- use
  - use(directives)
  - use(connos)

PREDICATES	NOMINALISATIONS
accept	
activate	
adjust	
affect	
allocate	
assume	
	attempt
block	blocking
	breaking
	calculation
call	
change	change
close	closing
come from	
command	command
concern	

connect	connection
consist	
	consumption
contain	
continue	
control	control
deexit	
deload	deloading
demand	
depend	
describe	description
detect	
disable	
disconnect	disconnection
do	
download	
drop	
enable	
exceed	
exit	
explain	
fail	
feed	
	flow
follow	
gate	
generate	
get	
give	
handle	
happen	
have	
ignore	
include	
indicate	indication
	input
	interface
keep track	
know	
latch	
lead	
limit	
list	
	loadsharing
lock	
maintain	
mean	
measure	
mention	
monitor	
occur	
open	
operate	operation
	order
	output
override	
perform	
prepare	
process	
produce	production

raise	
reach	
receive	
release	
remove	
	request
require	requirement
reservate	reservation
restart	
result	
run	
satisfy	
say	
scale	
secure	
select	selection
send	
show	
start	start, starting
stop	stop, stopping
succeed	
supervise	
switch	switching
synchronize	synchronization
take	
take place	
transfer	
transmit	
treat	
update	
use	use
vary	
want	

11 - Subordinate clauses

- nominalization + subord

- WHEN	2.1.14
- AFTER	2.1.26

- subord + main

- WHEN	2.1.3, 2.3.3, 2.3.6, 2.4.3, 2.4.5, 2.4.8, 2.4.9, 2.4.13, 2.4.15, 2.4.40, 2.5.7, 2.5.8, 2.5.10
- IF	2.3.14, 2.3.22, 2.3.26, 2.4.33, 2.4.48, 2.4.55
- IN CASE	2.3.12, 2.3.27
- AFTER	2.4.12

- main + subord

- WHEN	2.2.1, 2.4.41

- IF	2.2.5, 2.4.25
- SO THAT	2.3.25
- BEFORE	2.4.51
- WHILE	2.5.4

- subord + main + subord

- IF-WHEN	2.2.3, 2.2.4
- IF-SO THAT	2.3.23
- WHEN-UNTIL	2.5.5

Nominalization followed by a subordinate clause can be included in the category MAIN +SUBORD. In the case of complex sentences with SUBORD + MAIN + SUBORD, we consider the first conjunction as belonging to SUBORD + MAIN and the second one to MAIN + SUBORD.

CONJUNCTION	M+S	S+M
WHEN	5	12
IF	2	9
IN CASE		2
SO THAT	2	
AFTER	1	1
BEFORE	1	
WHILE	1	
UNTIL	1	

APPENDIX B4: Cohesion

1 - REFERENCE

See Appendix B2.

2 - COMPARISON

highest	2.3.10, 2.3.19
higher	2.3.24, 2.3.25
greater than	2.5.11
too	2.4.12
also	2.3.21
respectively	2.3.20

3 - SUBSTITUTION

one	2.3.24, 2.4.5, 2.4.33
partitive	2.2.2, 2.4.33, 2.4.48, 2.4.55
do	2.3.14, 2.3.15, 2.3.22, 2.3.23

4 - ELLIPSIS

- in a sentence

2.2.1	*is in AUTO-mode and (is) not blocked*
2.2.3	*the highest prioritied*
	BAD : *the one with the highest priority*
2.2.4	*the first (DG) in the standby sequence*
2.2.5	*if the former fails to start or switch online*
2.2.8	*this is either default or selected*
2.3.8	*which DGs are online and stopped*
2.3.19	*is always online and master DG*
2.3.20	*respectively*
2.3.24	*i.e. (will) start a new DG and then stop*
2.3.28	*is stopped and (is) blocked*
2.4.1	*to either B or BT/ST*
2.4.18	*can connect/disconnect*
2.4.23	*are disconnected and (are) stopped*
2.4.34	*can connect and (can) disconnect*
2.4.49	*to BT respectively (to) ST*
2.4.52	*is fully automatic and (is) controlled*
2.5.6	*is started and (is) switched online*

- between sentences

1.1.3	*the DG part of the system*
1.1.4	*the SG part*

Nominal ellipsis		2.2.3, 2.2.4
Verbal ellipsis		
	be	2.2.1, 2.2.8, 2.3.8, 2.3.19,
2.3.20,		
		2.3.28, 2.4.23, 2.4.52, 2.5.6
	will	2.2.4, 2.3.24
	can	2.4.18, 2.4.34
Preposition		

to 2.2.5, 2.4.1, 2.4.49

5 - Conjunctions

6 - Semantic relations

a - NOUNS

action
alarm
attempt
blackout
blocking
breaking
busbar
case
change
closing
command
component
condition
connection
consoles
consumer
consumption
control
current
deloading
description
diesel
disconnection
electricity
engine
failure
figure
frequency
function
generator
information
level
load
loadsharing
machinery
mode
operation
operator
overview
part
pitch
plant
position
possibility
power
power up
pressure
prewarning
priority
procedure
production
purpose

range
request
requirement
reservation
responsibility
scope
section
selector
sequence
ship
shutdown
signal
situation
slowdown
speed
start
starting
stop
stopping
strategy
switch
switching
synchronization
synchronizing
system
thruster
time
use
variation
voltage
way
word

CATEGORIES

Components

busbar
component
consumer
console
engine
generator
machinery
operator
plant
selector
ship
switch
system
thruster

Data/Information

alarm
condition
current
electricity
failure

frequency
information
level
load
mode
part
pitch
position
possibility
power
pressure
prewarning
priority
procedure
range
request
requirement
responsibility
sequence
signal
speed
strategy
variation
voltage

Action

action
attempt
blackout
blocking
breaking
change
closing
command
connection
consumption
control
deloading
disconnection
function
loadsharing
operation
power up
production
reservation
shutdown
slowdown
start/starting
stop/stopping
switching
synchronization/synchronizing
use

Text

case
description
figure
overview

purpose
scope
section
situation
word
way

b - Synonyms

We first had a look in the dictionary at the entries corresponding to the predicates. Then we listed all possible synonyms and then checked whether these synonyms were present in the text.

List of synonyms in the dictionary

accept
　　receive
　　approve
　　endure
　　understand
　　assume
　　undertake
activate
　　vitalize
adjust
　　settle, resolve
　　adapt
　　regulate
affect
　　fancy, cultivate, feign, pretend
　　frequent, incline, assume
　　influence, touch, impress, strike, sway
allocate
　　distribute, allot
　　designate
assume
　　receive, undertake, don, seize, usurp, feign, pretend
　　suppose
　　affect, simulate, counterfeit, sham
block
　　hinder, interfere, prevent, prohibit, limit
call

change
　　transform, alter, modify, vary
　　transfer
close
　　bar, block
　　end, conclude, terminate, complete, finish
come from

command
　　order, bid, enjoin, instruct, charge, direct, govern
　　demand, exact
　　ANT　*comply, obey*
concern
　　relate
　　involve, engage, occupy, matter
connect
　　join, fasten

ANT disconnect

consist

lie, reside

be made of

contain

restrain, control, check, halt

hold, comprise, include, enclose, bound

continue

maintain, remain

endure, stay, prolong, retain

last, abide, persist

control

check, test, verify

regulate

rule, conduct

deexit NO

deload NO

demand

ask, claim, summon, require, exact

depend

rely

describe

represent, delineate

detect

discover, determine, demodulate

disable

weaken, deprive

ANT rehabilitate

disconnect

ANT connect

do

download

NO see unload ?

drop

fall, reduce, unload, dismiss

enable

empower

exceed

extend, surpass, transcend, excel, outdo, outstrip, predominate

exit

NOT A VERB

explain

expound, explicate, elucidate, interpret

fail

weaken

miss, lack, neglect

feed

satisfy, gratify, support, encourage

supply

move into a machine

follow

succeed, ensue, supervene

ANT precede

gate

SPEC

generate

procreate, beget

produce (electricity)

get

obtain, procure, secure, acquire
gain, win, earn

give
present, donate, bestow, confer, afford
profer, allot, produce
bear, sell, deliver

handle
manipulate, wield
treat, manage, direct, touch

happen
chance, occur, transpire

have
hold, own, possess, enjoy

ignore
reject, neglect
ANT heed, acknowledge

include
enclose, comprehend, embrace, involve
ANT exclude

indicate
point out/to, demonstrate, suggest

keep track

know
believe, think, recognize, discern

latch
SPEC

lead
guide, direct
begin, open
ANT follow

limit
restrict, circumscribe, confine, prescribe

list
enumerate

lock
fasten, hold, bind

maintain
keep in state, sustain, continue, preserve, carry on, keep up
support, provide for
assert, defend, vindicate, justify

mean
intend, show, indicate
signify

measure
regulate, govern
estimate, appraise

mention
name, instance, specify
refer to

monitor
check, test, watch, observe, control
keep track of, regulate

occur
appear, take place, come to mind, happen

open

operate
perform, produce, effect, work

override

trample, dominate, annul, neutralize, overlap

perform

fulfill, carry out, do, act, function
play, execute, discharge, accomplish, achieve

prepare

make ready, ready
fit, qualify, condition, compound

process

prosecute
work, treat

produce

exhibit, yield, present
make, manufacture
accrue, bear, make

raise

awaken, arouse, incite
elevate, heighten

reach

gain, compass, achieve, attain

receive

accept, admit, take, acquire

release

free, relieve, relinquish, give permission

remove

change, transfer, move, dismiss, eliminate

require

claim, ask, call, demand, request, lack, impose

reservate

NO

restart

start anew, resume

result

proceed, arise as consequence
have an issue
revert

run

function, operate

satisfy

discharge, indemnify
please, convince, dispel, conform to, make true, be adequate
suffice
fulfill, meet, answer

say

express, state, utter, pronounce, recite, repeat, speak
indicate, show, communicate

scale

weigh in scale

secure

guarantee, ensure
effect, get, release

select

pick out, choose

send

deliver

show

manifest, evidence, demonstrate, exhibit

start

stop

succeed
come next, follow
ANT precede
thrive, flourish
ANT fail, attempt
supervise
superintend, oversee
switch
operate a switch
synchronize
SPEC
take
take place

transfer
convey, transport, transmit
transform, change
transmit
send, transfer, forward, convey, conduct
treat
deal, handle
update
bring up to date
use
employ, utilize
vary
change, diversify
deviate, depart
want
lack, require, desire

List of synonyms in the text

accept
receive
assume
affect
assume
assume
receive
affect
block
limit
change
vary
transfer
close
block
contain
control
include
continue
maintain
demand
require
drop
unload
feed

satisfy

follow

succeed

generate

produce

get

secure

give

produce

happen

occur

lead

open

maintain

continue

mean

show

indicate

monitor

control

keep track of

occur

happen

operate

perform

produce

receive

accept

take

remove

change

transfer

require

call

run

operate

say

indicate

show

succeed

follow

transfer

change

transmit

send

transfer

vary

change

want

require

LIST OF ANTONYMS

activate/release

block

remove/change/vary

continue/maintain

close/lock
    open
connect
    disconnect
enable
    disable
fail
    succeed
follow
    lead
mention
    ignore
start
    stop

APPENDIX B5: COMMENTS : PROBLEMS AND POSSIBLE SOLUTIONS

1.1.3 repetition of *system*
pb with *part*

the system -> *the PMS*

1.1.4 pb with *part*
repetition of *SG*

no solution

1.2 relation *components/system*
same entity or different entities ?
different point of view?

1.2.2 repetition of *GS*
MB
cannot be avoided, different entities

1.2.3 repetition of *alarm*
solution make clear there are different kinds of alarms

the PMS monitors the alarms sent by the alarm system,
(as well as)
the alarms detected by the PMS and the information coming from the machinery surrounding the DGs.

2.1.1 repetition of *three modes of operation*
cooccurrence DGs and SG -> GSs

problem are these modes the same or not ?
solution substitution or ellipsis
explicitation of identity

1 - the PMS contains three modes of operation for the DGs and three modes for the SG.
2 - the PMS contains three modes of operation for the DGs and three for the SG.
3 - the PMS contains three modes of operation for the DGs and the SG.
4 - the PMS contains three modes of operation for the GSs.
5 - the PMS contains three modes of operation for each GS.
6 - each GS has three modes of operation.
7 - the GSs have each three modes of operation.
8 - the PMS contains the same three modes of operation for each GS.

2.1.3 repetition of *mode*
solution substitution or ellipsis

1 - when MANUAL mode is selected it overrides the other two.
2 - when MANUAL mode is selected it overrides the others.
3 - when MANUAL mode is selected it overrides all
the
others.

2.1.5
2.1.6 repetition of *MB*
use of *in question*

solution different presentation
avoid the use of *in question*

SG no control of MBs whether to thrusters or
BB

2.1.7
2.1.8
2.1.9 repetition of *mode*
repetition of *DGs*
solution susbtitution or ellipsis
but problem : *modes* and *DGs* are plural ->
ambiguity
if pronouns for both.

1 - substitution of *mode*

They are common to all DGs.
(should be said before by the way)
They require that the DGs are in AUTO (not
MANUAL).
(easy to suppress mode with AUTO or MANUAL)

2 - substitution of *DG*

The next two modes concern operation of DGs.
These modes are common to all of them.

or

The next two modes concern operation of DGs.
They are the same for all of them.

2.1.16 repetition of *function*
solution susbtitution or ellipsis

1 - the PMS will perform the functions 1-4 described for ... and the following ones.
2 - the PMS will perform the functions 1-4 described for ... and the following.

2.1.18 repetition of *DG*
solution

1 - change to the next DG if one does not start.
2 - change to the next DG if the former one does not
start.
3 - change to the next DG if the former does not start.

4 - change to the next one if a DG does not start.
5 - change to the next if a DG does not start.
Cataphoric : not so bad.

6 - if a DG does not start change to the next.
Pb : list of functions -> better to start with the function and keep the same structure for all the elements of the list.

2.1.19 repetition of *DG*
solution

1 - start of standby DG and shutdown of faulty one

2.1.20 repetition of *SG*
mode

solution use of *mode* in brackets not necessary
1 - stopped either because its mode is changed to one without SG on the ship handling mode selector... (bad : it's not the SG mode, it's the ship mode)
2 - ... either because ship mode is changed to one without SG...

2.1.21 repetition of *start*
SG
solution the starts are different starts so it's OK

if SG has a standby start ... or if its frequency ...

2.1.22
2.1.23 repetition of *operation of*
difficult to avoid

2.1.30 see 2.1.16

2.2.1 not so good
solution
1 - "blocked" means that a DG is not available...
2 - a blocked DG is a DG which is not available...
3 - a DG is blocked when it is not available...

2.2.5 repetition of *DG*
solution
1 - if the former one...
2 - if the former...

2.2.3
2.2.5 repetition of *sequence*
2.2.7 different terms but same entity
2.2.10 *standby sequence*
master/standby sequence
priority sequence

2.2.7 repetition of *DG*

2.2.12 repetition of *standby sequence of DGs*

2.3.8
2.3.9 repetition of solution — *operator*

The operator controls ... He controls...

By the way both sentences are not so good :

1 - From the ISC consoles the operator controls the DGs which are online and stopped. In other words he controls the available power.
2 - From the ISC consoles the operator controls the online and stopped DGs. In other words he controls the available power.
3 - The operator controls the online and stopped DGs from the ISC consoles. In other words he controls the available power.

2.3.23 repetition of — *DG*
priority
problem — same DGs ?
solution
1 - ... it can be done by changing its priority to a lower one.
-> if it's the same DG

2 - ... it can be done by changing its priority so that the next online DG gets a lower one.

2.3.25 repetition of — *DG*
priority
problem — same DGs ?

2.4.3 repetition of — *BB*
solution
1 - when the SG is connected to BB, its frequency ...
not so good : not the frequency of the SG

2.4.13 repetition of — *BT/ST*
solution — replace by thrusters
their

1 - when the switch is in MANUAL position, the
MBs

to the thrusters are controlled from the MSB.
This include control of SG voltage during their power up.
In fact not necessary to say 'to the thrusters'.

2 - when the switch is in MANUAL position, the
MBs

are controlled from the MSB.
This include control of SG voltage during the thrusters power up.

2.4.15 repetition of — *BT/ST*

2.4.24	repetition of	*DG*
2.4.29	repetition of solution	*frequency*
	1 -	the frequency of the SG is in a range near to BB normal one
	2 -	SG frequency ... to BB normal one
	3 -	SG frequency ... to the normal one of BB
	4 -	... to that of BB
2.4.35	repetition of solution	*BT/ST* replace by *thrusters*
2.4.41	repetition of solution	*thruster*
	1 -	thruster is ready for operation when its current is at idle level.
2.5.2	repetition of solution	*power consumer*
	repetition of	*start blocking*

APPENDIX B6: Thematic progression

PMS 1,2

Here is an analysis of the thematic progression.

1
(a) The PMS [T1a] monitors and controls electricity production via four GSs [R1a].
(b) The four GSs [T1b] include three DGs and one SG [R1b].

The T1b is absent and Rb is fused with Ra.
The thematic structure is therefore :
T1a -> R1(a,b)

2
(a) The SG [T2a] is connected to the ME [R2a].
(b) The SG [T2b] can produce power to ... [R2b].

T2b = T2a
There is a composition expressed by way of AND.
The structure is :
T2a -> R2a + T2b (=T2a) -> R2b

3 The DG part of the system [T3] is a ... system [R3].
T3 -> R3

4 The SG part [T4] includes ... [R4].
T4 -> R4

We do not decompose further. Otherwise we could consider that elements of R4 are the same ones than in R2.

1.2
1 In figure 1a is shown the c/m system.

Two solutions :
- The theme is the left most constituent, the last element is the rheme.
- The theme is the grammatical subject.
Anyway we can have the same representation since it is independent of order.
T1 -> R1

2 This [T2] includes ... [R2].
T2 -> R2

3 The PMS [T3] monitors ... [R3].

2.1

1 The PMS [T1] contains ... [R1].

We can consider that R1 = R'1 + R"1.

1' They [T1'] are explained below [R1'].

2 Each GS [T2] has a selector [R2].

3 When MANUAL mode is selected it overrides the two other modes.

Problem with subordinate clauses: in Danes there are only examples of relatives or completives. The verb and its completive belong to the rheme.

Several solutions :

1 - Can we consider that subordinate clauses belong to the rheme.
Problem when the S-clause is initial, we have a disjoint rheme.
T3 -> R'3 + R"3
2 - When S-clauses are initial they are thematic.
T3 -> R3
3 - Consider that S-clauses (not relatives and not completives) are composed, just like coordinate clauses.
T3a -> R3a + T3b (= T3a) -> R3b

7 The next two modes [T7] only concern ... [R7].

8 These modes [T8] are ... [R8].

9 The modes [T9] require that ... [R9].

10 The PMS [T10] will perform the following functions [R10]

In fact we can consider that R10 = R11 + R12 + R13 + R14
Problem with the subordinate in R14 : belongs to the rheme ?

15 Start and stop of DGs [T15], except during blackout start, is commanded... [R15].

Pb : does 'except...' belong to the rheme ?
T15 -> R15

16
(a) The PMS [T16a] will perform the functions ... [R16a]
(b) The funtions [T16b] are described... [R16b]

T16b is deleted.
R16b is fused with R16a.

Here too we can consider that R16 = R17 + R18 + R19 + R20 + R21.
So the structure is quite complicated.

T16a -> R16a (= R'16ab + R"16a)

22 The next two modes [T22] only concern operation of SG [R22].
23 Operation of DG [T23] is independent of selected mode SG SEMIAUTOMATIC and SG AUTOMATIC [R23].
24 The PMS [T24] will perform the following functions [R24].

Here we can consider that R24 = R25 + R26 + R27 + R28.

29 Start and stop of SG to either BB or BT/ST [T29] is commanded by the operator [R29].
30 The PMS [T30] will automatically perform the functions 1-4 described for SG SEMIAUTOMATIC mode and the following functions [R30].

31 We can consider that R31 is included in R30.

2.2

1 Blackout start [T1a] is enabled [R1a] when at least one DG [T1b] is in AUTO mode and not blocked [R1b].
Blocked [T'] means that the DG is not available... [R'].

2 One of two actions [T2] will take place after a blackout [R2].
3 If one or more DGs [T3a] is running [R3a]
the highest prioritied [T3b] will be switched online [R3b]
when its frequency [T3c] has reached a preset level [R3c].

4 If no DG [T4a] is running [R4a]
the first in the standby sequence [T4b] will be started [R4b]
when its frequency [T4c] has reached a preset level [R4c].

5 The next DG in the standby sequence [T5a] will be started [R5a]
if the former DG [T5b] fails to start or switch online [R5b].

6 Switch online [T6] means ... [R6].

7 To decide the master/standby sequence of the DGs [R7b]
each DG [T7] always has a priority [R7a].
8 This [T8] is either default or selected... [R8].
9 Priorities [T9] are ... [R9].
10 The priority sequence [T10] is used in the PMS control modes to

13 R10 = R10 + R11 + R12 + R13

14 The priority [T14] is used to select ... [T14].

2.3

1 The DGs [T1] can be controlled ... [R1].
2 Switching between the different control possibilities [T2] is done with a switch, named M/A, mounted in the MSB [R2].

Here fusion of
(a) the switch is named M/A
(b) the switch is mounted in the MSB
into R2.

3 When the M/A switch [T3a] is in MANUAL position [R3a]
the DG [T3b] is controlled ... [R3b].
4 Synchronizing, ... and speed/load control [T4] is done from the MSB [R4].
5 This [T5] is called MANUAL mode [R5].
6 When the MANUAL/AUTO switch [T6a] is in AUTO position [R6a]
the DG [T6b] is ... [R6b].
7 In this situation the basic control [T7] is performed from the ISC system [R7].
8 The operator [T8] controls from the ISC consoles which ... [R8].
9 The operator [T9] controls in other words the available power [R9].
10 The online PMS-controlled DG with highest priority [T10] is frequency controlled [R10].
11 This [T11] is called the master DG [R11].
12 In case a critical condition, which could lead to a shut down [T12a] occurs [R12a]
an alarm [T12b] will be indicated [R12b].
13 Loadsharing [T13] is performed between all online PMS-controlled DGs [R13].
14 If the operator [T14]a wants to stop ... [R14a]

this [T14b] can be done ... [R14b].
15 This [T15] cannot be done ... [R15].
16 Stopping [T16] means ... [R16].
17 In the same way start of a stopped DG [T17] can be done ... [R17].
18 Starting [T18] means ... [R18].
19 The DG with the highest priority, under PMS control and not blocked [T19] is always online and master DG [R19]
(if SG operation to BB [T19'] is not selected [R19']).
20 The followings DGs [T20] are started...automatically [R20]
all dependent on their priority and the actual power consumption.
21 Loadsharing of ... [T21] is also part of the AUTOMATIC mode [R21].
22 If a PMS controlled DG [T22] is wanted ... [R22], this [R22'] can be done ... [R22'].
23 If stop [T23] is wanted ...[R23], this [T23'] can be done ... [R23'].
24 The PMS [T24] will then ... [R24].
25 In the same way start of ... [T25] can be done... [R25]
so that the stopped DG [T25'] gets a higher priority [R25'].
26 If the PMS control mode [T26] is changed... [R26] the plant [T26'] will ... [R26'].
27 In case an alarm for standby start [T27] occurs [R27], a standby DG [T27'] is started [R27'].
28 Then the faulty DG [T28] is stopped and blocked [R28].

2.4 Shaft generator control [T]

1 The SG [T1] can connect to either BB or BT/ST [R1] and controls [T2] are performed either from the PMS or from the MSB [R2].
2 It [T3] is impossible to connect the SG to thrusters and to the BB at the same time [R3].
3 When the SG [T4] is connected to BB [R4] the BB frequency [T5] depends on the ME RPM [R5].
4 Switching between control possibilities [T6] is performed with a switch ... [R6].
5 When the M/A switch [T7] is in M-position [R7]
synchronization... [T8] are done from the MSB [R8].
6 This [T9] is called MANUAL mode [R9].

PMS 4,5

4.1
1 The PMS subsystem [T1] consists of two main blocks [R1]:
the DG control block [R2a] and the SG control block [R2b]
(general block description [T3] is in chapter 5 [R3]).

2 Each block [T4] is functional equal [R4],
i.e. the DG block [T5] is a ... [R5]
(the PMS [T6] handles... [R6])
and the SG block [T7] is ... [R7]
(the PMS [T8] handles ... [R8]).

3 Each GS block [T9] is ... [R9].

4 In this chapter these main blocks [T10]
will be described in details [R10].

5 Furthermore various exceptions as power up conditions [T11]
are described in details [R11].

4.2 DG block diagram [T]

1 The DG control block [T1] consists of ... [R1].
2 Module numbering [T2] follows ... [R2].
3 Below [T3] is a description and purpose of each module [R3].
4 If nothing [T4] is mentioned about the DG status [R4],
it [T5] is assumed that ... [R5].

1.1 Load control module [T]
1 This module [T1] has two functions [R1]:
it [T2] performs DG loadsharing [R2] and
it [T3] deloads and opens ... [R3].

2 Loadsharing [T4]
Independent of PMS mode
all DGs online [T5] have the same percentage load [R5].

3 The reference to the loadsharing [T6] is ... [R6].

4 Deload [T7]
During stop,
the DG [T8] is deloaded ... [R8].

5 Then the MB [T9] is opened [R9].

6 If deload [T10] is not performed... [R10],
a deload failure [T11] is ... [R11].

7 The LOAD CONTROL MODULE [T12] is ... [R12].

1.2 FREQUENCY CONTROL MODULE [T]

1 The master DG [T1] is frequency controlled from this module [R1]
except if a SG [T2] is online [R2].

2 This module [T3] is controlled from ... [R3].

1.3 MB ON CONTROL MODULE [T]

1 This module [T1] is used ... [R1].
2 MB ON [T2] can happen in two situations [R2] :
3 Normal synchronization [T3] :
BB frequency [T4] is used for synchronization [R4].
4 The DG frequency [T5] is raised [R5]
so the frequency [T6] is slightly above BB frequency [R6].
5 Then the synchronizer [T7] is enabled for synchronization [R7]
and the DG frequency [T8] is ... [R8].
6 If the synchronization [T9] succeeds [R9],
the MB [T10] is closed [R10]
and the synchronizer enable signal [T11] is removed [R11].
7 If no synchronization [T12] is performed [R12],
the synchronizer enable signal [T13] is removed [R13]
and a synchronization failure [T14] (is) generated [R14].
8 Blackout start [T15]:
The MB [T16] is closed [R16],
when DG frequency [T17] is near reference frequency [R17].

9 If the DG [T18] doesn't reach the BB frequency [R18],
if the DG [T19] fails to synchronize [R19]
or if the MB [T20] is not closed [R20]
a synchronize failure [T21] is send to ... [R21].
10 This module [T22] is ... [R22].

1.4 EXT. MB ON CONTROL MODULE [T]

1 This module [T1] is used ... [R1].
2 The module [T2] is enabled... [R2]
and the master DG [T3] will start ... [R3].
3 This means that the master DG [T4] ... [R4].
4 At the same time the synchronizer [T5] is enabled [R5].
5 This [T6] continues ... [R6].
6 If synchronization [T7] succeeds [R7]
or if no synchronization [T8] is performed [R8],
the external synchronizer [T9] is disabled [R9].
7 If the synchronization [T10] fails [R10],
an external synchronization failure [T11] is generated [R11].
8 The signal [T12] is removed [R12],
when the request for synchronization [T13] is removed [R13].
9 This module [T14] is controlled ... [R14].

1.5 PRELUB MODULE [T]
1 Optional [R1]

2.1 START/RUN/STOP MODULE [T]
1 This module [T1] starts resp. stops the DG [R1]
when a command for start resp. stop [T2] is received [R2].
2 The module [T3] controls the ... [R3]:
3 When the DG [T4] is not master and online [R4]
it [T5] activates loadsharing [R5].
4 When the DG [T6] synchronizes [R6],
it [T7] ... [R7].
5 When the DG [T8] is master [R8],
it [T9] ... [R9].
6 A start command under PMS control [T10] results... [R10].
7 If the start [T11] failed [R11],
a start failure [T12] ... [R12].
8 This module [T13] will ... [R13],
so the AE [T14] is stopped [R14].
9 Start commands [T15] are generated in the following modules [R15]:
10 3.1 MODULE [R15a] when a standby DG [T16] is ... [R16].
11 3.3 MODULE [R15b] when a blackout [T17] occurs [R17].
12 5.1 MODULE [R15c] when a DG [T18] is ... [R18].
13 3.2 MODULE [R15d]. When the SG [T19] wants to stop [R19],
it [T20] releases ... [R20]
(released if the DGs [T21] are ... [R21]).
14 3.4 MODULE [R15e] when an alarm for standby start of the DG [T22]
is detected [R22].
15 Stop commands [T23] are generated in the following modules [R23]:
16 3.1 MODULE [R23a] when the DG [T24] ... [R24].
17 3.2. MODULE [R23b] immediate stop of DG
if a SG [T25] is online [R25].
18 5.1 MODULE [R23c] when ... [R26] the DG [T27] is... [R27].
19 3.4 MODULE [R23d] when the AE [T28] has shutdown [R28],

MB-trip or AE failure .

3.1 DG AUTOMATIC MODE START/STOP MODULE [T]
1 This module [T1] will control that the master DG is online [R1]
unless a SG [T2] is connected to the BB [R2].
2 If the available power [T3] drops ... [R3]
or if the available power [T4] ... [R4],
the module [T5] ... [R5].
3 If no standby DG [T6] can start [R6]
or if no standby DG [T7] is available [R7],
this module [T8] ... [R8].
4 If the available power [T9] ... [R9],
this module [T10] generates... [R10].
5 Furthermore this module [T11] ... [R11]
ie if a stopped DG [T12] has ... [R12],
it [T13] ... [R13].

3.2 SG START/STOP/CONTROL MODULE [T]
1 Input frequency from the SG [T1] ... [R1].
2 If more than one SG [T2] is ... [R2],
the correct SG [T3] ... [R3]
and a synchronize error ... [T4] is gated ... [R4].
3 Upon receiving ... this module [T5] enables ... [R5],
if the resp. SG [T6] is online [R6].

3.3 BLACKOUT MODULE [T]
1 If all MBs to BB [T1] are open and there is ... [R1]
this module [T2] ... [R2]:
2 First, ... [R2a]
3 Second, ... [R2b]
4 The DG ... [T3] is ... [R3]
and in both cases the MB [T4] is closed [R4]
when the frequency [T5] ... [R5].
5 This module [T6] ... [R6].

3.4 FAILURE MODULE [T]

1 All PMS alarms/warnings [T1] are fed through this module [R1].
2 The alarms/warnings [T2] are latched [R2].
3 The following [T3] is ... [R3].
4 Standby start [T4]
If the DG [T5] is online [R5],
the mode [T6] is ... [R6]
and a standby start alarm [T7] is detected [R7],
this module [T8] ... [R8].
5 When the AE [T9] is stopped [R9],
it [T10] is blocked [R10].
6 In ... an alarm [T11] is generated [R11]
and in ... a warning [T12] is generated [R12].
7 In all modes a DG subgroup alarm [T13] ... [R13].
8 By use of ... [T14] it is ... [R14].
9 Shutdown [T15]
The AE [T16] ... [R16]
(the safety system [T17] ... [R17],
this stop [T18] ... [R18])

and when it [T19] ... [R19],
it [T20] ... [R20].
10 A DG subgroup alarm [T21] ... [R21].
11 By ... [T22] ... [R22].
12 Start fail [T23]
The AE [T24] ... [R24].
13 A DG subgroup alarm [T25] ... [R25].
14 MB fail [T26]
This [T27] includes the following failures ... [R27].
15 ... [R27a]
16 ... [R27b]
17 ... [R27c]
18 In ... the AE [T28] ... [R28].
19 A DG MB subgroup alarm [T29] ... [R29].
20 MB open/close [T30]
In ... a... warning [T31] ... [R31]
and a DG MB subgroup alarm [T32] ... [R32].
21 Deload fail [T33]
In... a deload failure ... [T34] ... [R34].
22 MSB fail [T35]
The following alarms [T36] ... [R36]
23 ... [R36a]
24 ... [R36b]
25 These failures [T37] ... [R37].
26 All failures [T38] ... [R38].
27 When the DG [T39] ... [R39],
a DG subgroup alarm [T40] ... [R40].

4.1 MASTER/STANDBY DECISION MODULE [T]
1 This module [T1] ... [R1] the DG [T2] ... [R2].
2 Furthermore it [T3] ... [R3]:
3 ... [R3a]
4 ... [R3b] and ... [R3c]
5 ... [R3d] and ... [R3e]
6 The priorities ... [T4] ... [R4]
ie if an offline DG [T5] ... [R5],
the offline DG [T6] ... [R6].
7 This module [T7] ... [R7].
8 This signal [T8] ... [R8].

4.2 POWER CALCULATION MODULE [T]
1 Calculation ... [T1] is performed in this module [R1].
2 Each GS alpha [T2] ... [R2].

5.1 DG MODE/COMMAND MODULE [T]
1 This module [T1] ... [R1]:
2 ... [R1a]
3 ... [R1b]
4 ... [R1c]
5 If the DG [T2] ... [R2],
no commands [T3] ... [R3].
6 If the DG [T4] ... [R4],
this module [T5] ... [R5].
7 This module [T6] ... [R6].

5.2 INTER ALPHA CONTACT MODULE [T]
1 This module [T1] ... [R1].
2 This [T2] ... [R2].
3 A watchdog clock [T3] ... [R3].
4 Furthermore this module [T4] ... [R4].
5 If relevant data ... [T5] is ... [R5],
the data ... [T6] ... [R6].
6 Values ... [T7] are transferred [R7],
when the alpha [T8] ... [R8].
7 The module [T9] ... [R9],
ie it [T10] ... [R10].

5.3 STATISTICS MODULE [T]
1 This module [T1] ... [R1].

6.1 COMMON/SPECIAL FUNCTION MODULE [T]
1 Optional

6.2 ANALOG INPUT MODULE [T]
1 This module [T1] ... [R1].
2 The analog inputs [T2] ... [R2].

6.3 DIGITAL I/O MODULE [T]
1 All digital i/o [T1] is fed through this module [R1].

2 Each i/o [T2] has ... [R2].

4.1

T	identity deletion

(1) T1 --> R1 : T2 --> R2a and R2b --> (T3 --> R3)

(2) T4 --> R4 ie T5 --> R5 (T6 --> R6) and T7 --> R7 (T8 --> R8)

(3) T9 --> R9

(4) T10 --> R10

(5) T11 --> R11
not described

§ 4.3 Exceptions	§ 4.3 SG x block here

4.2 **T**

(1) T1 --> R1

(2) T2 --> R2

(3) T3 --> R3

(4) if T4 --> R4 + T5 --> R5

R3 = HYPERTHEME

1.1 T

(1) T1 -> R1 : T2 -> R2 and T3 -> R3

2 possibilities

T1 -> R1a and R1b

(2) T4 : T5 -> R5

(3) T4 : T6 -> R6

T4 T7 = HYPERTHEMES	

(4) T7 : T8 -> R8
(5) T9 -> R9
(6) T10 -> R10 + T11 -> R11
(7) T12 -> R12

1.2 T

(1) T1 -> R1

(2) T2 -> R2

1.3 T

(1) T1 -> R1a and R1b

(2) T2 -> R2

T3 -> R3a and R3b

T4 T13 = HYPERTHEMES	

(3) T4

(4) T5 -> R5 + T6 -> R6

(5) T7 -> R7 and T8 -> R8

(6) T9 -> R9 + T10 -> R10 and T11 -> R11

(7) T12 -> R12 + T13 -> R13 and T14 -> R14

(8) T13
T15 -> R15 + T16 -> R16

(9) if T17 -> R17 or if T18 -> R18 or if T19 -> R19 + T20 + R20

(10) T21 -> R21

1.4 T

(1) T1 -> R1

(2) T2 -> R2 and T3 -> R3

(3) T4 -> R4a and R4b

T4 = HYPERTHEME	

(4) T5 -> R5

(5) T6 -> R6

(6) T7 -> R7 or T8 -> R8 + T9 -> R9

(7) T10 -> R10 + T11 -> R11

(8) T12 -> R12 + T13 -> R13

(9) T14 -> R14

1.5 T

(1) T1 -> R1

2.1 T

(1) T1 -> R1 + T2 -> R2

(2) T3 -> R3
T4 T4 = DELETED HYPERTHEME
(3) T5 -> R5 + T6 -> R6

(4) T7 -> R7 + T8 -> R8

(5) T9 -> R9 + T10 -> R10
T11 = T10a T23 = T10b

(6) T11 -> R11

(7) T12 -> R12 + T13 -> R13

(8) T14 -> R14 + T15 -> R15

(9) T16 -> R16

(10) R16a + T17 -> R17 SPLIT RHEME

(11) R16b + T18 -> R18

(12) R16c + T19 -> R19

(13) R16d + T20 -> R20 + T21 -> R21

(14) R16e + T22 -> R22

(15) T23 -> R23

(16) R23a + T24 -> R24 SPLIT RHEME

(17) R23b + T25 -> R25 + T26 + R26

(18) R23c + T27 -> R27 + T28 -> R28

(19) R23d + T29 -> R29 + T30 -> R30

3.1 **T**

(1) T1 –> R1 + T2 –> R2

(2) T3 –> R3 + T4 –> R4 + T5 –> R5 CONSTANT THEME

(3) if T6 –> R6 or if T7 –> R7 + T8 –> R8

(4) T9 –> R9 + T10 –> R10

(5) T11 –> R11 ie T12 –> R12 + T13 –> R13

3.2 **T**

(1) T1 –> R1 R1 = HYPERTHEME

(2) T2 –> R2 + T3 –> R3 and T4 –> R4

(3) T5 –> R5 + T6 –> R6

3.3 **T**

(1) if T1 –> R1 and if T2 –> R2, T3 –> R3 R3 = SPLIT RHEME

(2) R3a

(3) R3b

(4) T4 –> R4 and T5 –> R5 when T6 –> R6

(5) T7 –> R7

3.4 T

(1) T1 -> R1

(2) T2 -> R2

R3 = SPLIT RHEME	

(3) T3 -> R3

(4) T4 (= R3a)
if T5 -> R5, T6 -> R6 and T7 -> R7 + T8 -> R8a and R8b

(5) when T9 -> R9, T10 -> R10
(6) T11 -> R11 and T12 -> R12
(7) T13 -> R13
(8) T14 -> R14
(9) T15 (= R3b)
T16 -> R16 (T17 -> R17, T18 -> R18) and when T19 -> R19, T20 -> R20

(10) T21 -> R21

(11) T22 -> R22

(12) T23 (= R3c)

(13) T24 -> R24

T25 -> R25

(14) T26 (= R3d)

T4 T15 T23 T26 = HYPERTHEMES T30 T33 T35	

T27 -> R27

(15) R27a (16) R27b (17) R27c

(18) T28 -> R28a and R28b

(19) T29 -> R29

(20) T30 (= R3e)

T31 -> R31 and T32 -> R32

(21) T33 (= R3f)

T34 -> R34

(22) T35 (= R3g)

T36 --> R36

(23) R36a (24) R36b (25) T37 --> R37

(26) T38 --> R38

(27) T39 --> R39

4.1 T

(1) T1 --> R1 to T2 --> R2

(2) T3 --> R3

(3) R3a (4) R3b and R3c (R3c') (5) R3d and R3e (R3e')

(6) T4a and T4b --> R4 ie if T5 --> R5, T6 --> R6 and T7 --> R7

(7) T8 --> R8

(8) T9 --> R9

4.2 T

(1) T1 --> R1

(2) T2 --> R2

5.1 T

(1) T1 --> R1

(2) R1a (3) R1b (4) R1c

(5) if T2 --> R2, T3 --> R3

(6) if T4 --> R4, T5 --> R5

(7) T6 --> R6

5.2 T

(1) T1 -> R1

(2) T2 -> R2

(3) T3 -> R3

(4) T4 -> R4

(5) if T5 -> R5, T6 -> R6

(6) T7 -> R7, when T8 -> R8

(7) T9 -> R9 ie T10 -> R10

5.3 T

(1) T1 -> R1

6.1 T

(1) T1 -> R1

6.2 T

(1) T1 -> R1

(2) T2 -> R2

6.3 T

(1) T1 -> R1

(2) T2 -> R2

APPENDIX B7: Topics and themes

1.1.1 level1-functions(component)
level1-structure(component)

(a) The PMS [T1a] monitors and controls electricity production via four GSs [R1a].
(b) The four GSs [T1b] include three DGs and one SG [R1b].

Here we have two different analyses:
- two topics
- two thematic structures fused into one
and the separation is not at the same place.

Alternative:

(1) The PMS [T1] monitors and controls electricity production [R1].
(2) Electricity production [T2] is performed via four GSs [R2].
(3) The four GSs [T3] include three DGs and one SG [R3].

TP: T1 -> R1 + T2 -> R2 + T3 -> R3

- T2 is deleted
- R1 and R2 are fused
- T3 is deleted
- R3 is fused with R1+R2
-> complex R
TP: T -> R1,R2+R3

Topics:
(1) level1-functions(PMS)
(2) level1-structure(PMS)
(3) list-of-elements(GSs)

Thus, the relation between the TP and the topics is the following:

```
(1)   T ->  C
      |     |
      T1    R1

(2)   T ->  C
            |
            R2

(3)   T ->  C
            |
            R3
```

T1 -> R1,R2+R3

1.1.2 level1-structure(SG)
level1-functions(SG)

(a) The SG [T2a] is connected to the ME [R2a].
(b) The SG [T2b] can produce power to ... [R2b].

T2b = T2a
There is no deletion of the second theme (it); two utterances are composed with AND.

The structure is :

T2a -> R2a and T2b (= T2a) -> R2b

In both cases:
topic = theme
comment = rheme

1.1.3 level2-functions(DG)

The DG part of the system [T3] is a ... system [R3].
T3 -> R3

topic = theme
comment = rheme

1.1.4 level2-functions(SG)

The SG part [T4] includes ... [R4].
T4 -> R4

topic = theme
comment = rheme

1.2 structural-overview(PMS)

1.2.1 ref-to(figure)

In figure 1a is shown the c/m system.

Two solutions :
(1) The theme is the left most constituent, the last element is the rheme.
(2) The theme is the grammatical subject.
Anyway we can have the same representation since it is independent of order.
T1 -> R1

(1) topic = theme
comment = rheme

1.2.2 list-of-components(PMS)

This [T2] includes ... [R2].
T2 -> R2

topic = theme
comment = rheme

1.2.3 level1-functions(PMS) -> should be in 1.1

The PMS [T3] monitors ... [R3].

topic = theme
comment = rheme

2 functional-description(PMS)
2.1 overview(control)

2.1.1 number(control-modes)

The PMS [T1] contains ... [R1].

We can consider that R1 = R'1 + R"1.

Here, the rheme is more informative than the topic suggested: not only the number of control modes but the beneficiaries of these modes.

Since the correspondence theme/first-argument of the topic seems quite regular, the topic could be: number(PMS,control-modes) or more general, number(component,function-modes). If the question is "How many control modes has the PMS?", the answer, i.e. the rheme, would be THREE.

We are now quite close to what Iordanskaja suggested.
There are three relevant elements:
- the topic corresponding to the theme,
- the parameter of interest corresponding to the predicate,
- the comment corresponding to the rheme.
This correspondence is quite clear in simple cases.

To be more precise, it is the first argument of the topic which corresponds to the theme.

For instance:

functions -> the predicate is the function itself, no "verbe support"
structure -> "via"/"connected to" (-> list-of-connections)
list-of-components -> "include"

2.1.1'

They [T1'] are explained below [R1'].

2.1.2 command-device(control-mode)

Each GS [T2] has a selector [R2].

Where does the theme (GS) come from? Should it be an argument of the topic?

argument = ?
predicate = rheme
comment = theme ?

2.1.3 effect(device)

When MANUAL mode is selected it overrides the two other modes.

Is the topic "effect" always realized as a conditional ?

2.1.4 description(control,DG,manual-mode,PMS)
2.1.5 description(control,SG,manual-mode,PMS)
2.1.6

Nominalisations with their arguments.
The agent (the PMS) is not realized.

2.1.7 description(auto-mode,DG)
2.1.8
2.1.9

The next two modes [T7] only concern ... [R7].

These modes [T8] are ... [R8].
The modes [T9] require that ... [R9].

topic = theme
2d arg included in rheme

2.1.10 actions(agent,mode,patient)

The PMS [T10] will perform the following functions [R10].

agent = PMS
agent = theme
topic -> "perform functions"
comment = rheme

Problem with the subordinate in R14 : belongs to the rheme ?

2.1.15 actions(agent,mode,patient)

Start and stop of DGs [T15], except during blackout start,
is commanded... [R15].

Pb : does 'except...' belong to the rheme ?

agent = operator

Quite interesting: it is the same topic, but the agent is different, and the thematic structure of the sentence is different from 2.1.10. 2.1.10: agent as theme + functions as rheme, active verb2.1.15: function as theme + agent as rheme, passive verb.It seems that the action, following the list of actions in 11-14, has to be at the beginning of the sentence. The focus is on the agent. So the function, which should be the comment, is thematized.Is cohesion better this way?

2.1.16 actions(agent,mode,patient)

(a) The PMS [T16a] will perform the functions ... [R16a]
(b) The funtions [T16b] are described... [R16b]

2.2 operation(PMS)

2.2.1 conditions(blackout-start,DG)

Blackout start [T1a] is enabled [R1a] when
at least one DG [T1b] is in AUTO mode and not blocked [R1b].

1st arg = theme
topic -> "is enabled when"
comment = rheme

definition(blocked,DG)
Blocked [T] means that the DG is not available... [R'].

1st arg = theme
definition -> "means that"
comment = rheme

Should DG be an argument?

2.2.2 level1-actions(blackout-start,DG)

One of two actions [T2] will take place after a blackout [R2].

1st arg : at the end of the rheme; could be thematized
actions -> theme + "take place"
comment -> 2.2.3 & 2.2.4: conditionals

Alternative: "After a blackout one of the following actions will take place:"

2.2.6 definition(switch-online,DG)

Switch online [T6] means ... [R6].
Same as 2.2.1

2.2.7 conditions(priority-decision,DG)

To decide the master/standby sequence of the DGs [R7b]
each DG [T7] always has a priority [R7a].

1st arg = theme
conditions -> infinitive introduced by "to", or "in order to", at the beginning of the sentence.
comment = rheme

2.2.8

This [T8] is either default or selected... [R8].
The sentence is not so good.

topic: agent(priority-decision) ?

2.2.9 topic: list-of-elements(priorities) ?

Priorities [T9] are ... [R9].

1st arg = theme
list-of -> "are"
comment = rheme

We could also have as topic : number(priorities).

2.2.10 level1-actions(priority-decision,DG)

The priority sequence [T10] is used in the PMS control modes to

1st arg = theme
actions -> "is used to"
comment = rheme

2.2.14

The priority [T14] is used to select ... [T14].
Same as before.

The topic is not SUMMARY.
The theme should be "In case of blackout...".

2.3.1 level1-description(control,DG)

The DGs [T1] can be controlled ... [R1].

2d arg = theme
topic + 1st arg: "can be controlled"
comment = rheme

2.3.2 command-device(DG)

Switching between the different control possibilities [T2]
is done with a switch, named M/A, mounted in the MSB [R2].

Here fusion of
(a) the switch is named M/A
(b) the switch is mounted in the MSB
into R2.

Alternative:
(1) switching ...is done with a switch -> effect(device)
(2) the switch is named... -> name(device)
(3) the switch is mounted in the MSB -> location(device)

effect(device):
- action performed by device in general
- action performed under certain conditions
two possibilities
- device = theme => a switch... helps to...
- effect = theme => switching is done with a switch

2.3.3 agents(control,manual-mode,DG)

When the M/A switch [T3a] is in MANUAL position [R3a]
the DG [T3b] is controlled ... [R3b].

Shouldn't the device be mentioned as argument?
Isn't it part of the effect of the device?

1st arg (device) = theme of cond
2d arg (M-mode) = rheme of cond
3d arg (DG) = theme of main
control -> "is controlled"
agents = rheme of main

2.3.4 level1-actions(control,manual-mode,DG)

Synchronizing, ... and speed/load control [T4]
is done from the MSB [R4].

The comment is thematized with the actions expressed as nominalizations.
The agent is part of the rheme.
2d and 3rd args: implicit

2.3.5
This [T5] is called MANUAL mode [R5].

I suggest the topic: denomination(object) -> term

2.3.6 agents(control,auto-mode,DG)
When the MANUAL/AUTO switch [T6a] is in AUTO position [R6a]
the DG [T6b] is ... [R6b].

cf 2.3.3

2.3.7

In this situation the basic control [T7] is performed
from the ISC system [R7].

Maybe there should a distinction between
- the agent of the control: e.g. PMS or operator
- the device used to control: e.g. ISC consoles, MSB, ...

2.3.8 agents(control,mode,DG)
functions(control,mode,DG,operator)

The operator [T8] controls from the ISC consoles which ... [R8].

(1) the operator controls which DG... -> function
(2) this control is performed from the ISC consoles -> device

2.3.9

The operator [T9] controls in other words the available power [R9].

agent is the theme -> it should be the first argument
control -> predicate (active form)
comment = rheme

2.3.10 functions(control,mode,DG,PMS)

The online PMS-controlled DG with highest priority [T10]
is frequency controlled [R10].

PMS: part of the theme?
DG -> theme: because of the absence of agent?

2.3.11

This [T11] is called the master DG [R11].

topic: denomination
passive vs active : "it is called" vs "we call it"
the term is rhematic: new information

2.3.12 alarm-rules(DG,mode)

In case a critical condition, which could lead to a shut down [T12a]
occurs [R12a]
an alarm [T12b] will be indicated [R12b].

2.3.13 level1-actions(agent,object,mode)

Loadsharing [T13] is performed between all online PMS-controlled
DGs [R13].

comment = theme
action -> "is performed"
agent: part of rheme ?
2d arg: part of rheme

2.3.14

If the operator [T14]a wants to stop ... [R14a]
this [T14b] can be done ... [R14b].

agent = theme

comment = rheme

2.3.15
This [T15] cannot be done ... [R15].

2.3.16 definition
Stopping [T16] means ... [R16].

2.3.17 level1-actions
In the same way start of a stopped DG [T17] can be done ... [R17].

comment = theme (nominalization)
in the same way -> cohesion
ISC consoles -> device -> agent

2.3.18 definition
Starting [T18] means ... [R18].

2.3.19 functions(function,mode,object,agent)

The DG with the highest priority, under PMS control and not
blocked [T19] is always online and master DG [R19]
(if SG operation to BB [T19'] is not selected [R19']).

object = theme
Is it really a function?

2.3.20
The followings DGs [T20] are started...automatically [R20]
all dependent on their priority and the actual power consumption.

object = theme
comment = rheme

2.3.21 level1-actions(agent,object,mode)

Loadsharing of ... [T21] is also part of the AUTOMATIC mode [R21].

comment = theme
object = part of theme
mode = rheme

2.3.22
If a PMS controlled DG [T22] is wanted ... [R22],
this [R22'] can be done ... [R22'].

object = theme
action = rheme
this = cf action

2.4 Shaft generator control

2.4.1 overview(control,SG)
The SG [T1a] can connect to either BB or BT/ST [R1a]

Here the control function does not seem relevant. The structure and
the connections of the SG are mentioned again.

agents(control,SG)

and controls [T1b] are performed either from the PMS
or from the MSB [T1b].

1st arg = theme
agents -> "are performed"
comment = rheme

2.4.2
It is impossible to connect the SG to thrusters and
to the BB at the same time.

2.4.3
When the SG [T3a] is connected to BB [R3a]
the BB frequency [T3b] depends on the ME RPM [R3b].

2.4.4 command-device(object)
Switching between control possibilities [T4]
is performed with a switch ... [R4].

object: not mentioned
command: theme
device-> comment = rheme
Quite different from the other command-device. Should it be effect(device) ?

2.4.5 level1-actions(function,mode,object,object)
When the M/A switch [T5a] is in M-position [R5a]
synchronization... [T5b] are done from the MSB [R5b].

device= theme1
position= rheme1
actions= theme2
agent=rheme2

2.4.6
This [T6] is called MANUAL mode [R6].
denomination(object)

APPENDIX C

APPENDIX C1: TEXT UNITS NUMBERING

Text units are typographical units. A text unit is a sequence :

$x\ a_1 \ldots a_n\ y$

where :

a_i is a typographic character which is not ":" or ".".
x is a capital letter preceded by a number or "." or ":" or, in one special case (see below), by "(".
y is "." or ":".

Special case :

(w)

is considered a text unit when w is an independent full sentence (cf. text unit 2.2.1).

Letters are used to distinguish text units in the same line.

Example :

2.1.4a DG's : 2.1.4b No control at all of DG in question.

In certain cases (titles and pseudotitles, see 1.2.2) a decimal is used to distinguish text units.

Example :

2.1.9 The models require...
2.1.9.1 DG SEMIAUTOMATIC :

APPENDIX C2: Analytical table

T Unit	Type Phrase	Verb Forms							Coordination		N (/)	N (E)
		N	V1	V2	V3	V4	V5	V6	N			
1	T											
1.1	T								1	C1		
1.1.1	Co-S	3	3						3	C2 C8 (2)		
1.1.2	Co-S	2		1	1				2	C7 C8	1	
1.1.3	S	1	1						1	C4	1	
1.1.4	S	1	1						1	C3	1	
1.2	T										1	
1.2.1	S	1		1							1	
1.2.2	S	1	1						1	C3	1	
1.2.3	S	1	1						1	C7		
2	T											
2.1	T											
2.1.1	Co-S	2	1	1					2	C3 C8		
2.1.1.1	T:											
2.1.2	S	1	1								1	
2.1.3	Sub-S1	2	1	1					1	C5	1	
2.1.4a	T:											
2.1.4b	St											
2.1.5a	T:											
2.1.5b	T:								1	C7		
2.1.5c	St											
2.1.6a	T:											
2.1.6b	St											
2.1.7	S	1	1									
2.1.8	S	1	1									
2.1.9	cS-+	2	2									
2.1.9.1	T:											
2.1.10	S	1			1							
2.1.11	St											
2.1.12	St								1	C1		
2.1.13	St											
2.1.14	Sub-St	1		1								
2.1.15	S	1		1					1	C1		
2.1.15.1	T:											
2.1.16	S	1			1				1	C3		
2.1.17	St								1	C1		
2.1.18	Sub-St	1						1				
2.1.19	St								1	C3		
2.1.20	Sub-St	2		2					2	C5 C9		
2.1.21	Sub-St	2	2						1	C8	1	
2.1.22	S	1	1									
2.1.23	S	1	1						1	C5		
2.1.23.1	T:											
2.1.24	S	1			1							
2.1.25	St											

2.1.26	Sub-St	1		1								
2.1.27	St											
2.1.28	St								1	C7		
2.1.29	S	1		1					2	C1 C7	1	
2.1.29.1	T:											
2.1.30	S	1			1				1	C3		
2.1.31	St								1	C7	1	
2.2	T											
2.2.1a	Sub-S1	3	1	2					1	C2		
2.2.1b	cS-+		2									
2.2.2	S	1			1							
2.2.3	Sub-S5	3				1		2				
2.2.4	Sub-S5	4				2		2	1	C8		
2.2.5	Sub-S2	4	1			1	2		1	C8		
2.2.6	S	1	1									
2.2.7	Sub-S4	2	1								1	
2.2.8	cS+-	2	1	1					1	C2		
2.2.9	S	1	1									
2.2.9.1	St											
2.2.9.2	St											
2.2.9.3	St											
2.2.10	S	1		1								
2.2.11a	T:											
2.2.11b	cS-+	2	1				1					
2.2.12a	T:											
2.2.12b	cS-+	3		1			2		1	C3		
2.2.13	cS-+	2		1			1					
2.2.14	cS-+	3		2			1					
2.3	T											
2.3.1	S	1				1			1	C6	1	
2.3.2	S	1		1							1	
2.3.3	Sub-S1	2	1	1					1	C9	1	
2.3.4	S	1		1					1	C9	2	
2.3.5	S	1		1								
2.3.6	Sub-S1	2	1	1							1	
2.3.7	S	1		1								
2.3.7.1	T											
2.3.8	cS-+	3	2	1					1	C2		
2.3.9	S	1	1									
2.3.10	S	1	1						1	C4		
2.3.11	S	1		1								
2.3.12	Sub-S5	3	1			1		1				
2.3.13	S	1		1					1	C4		
2.3.14	Sub-S2	3	1			1	1		1	C4		
2.3.15a	T	1				1						
2.3.15b	S											
2.3.16	S	1	1						1	C2		
2.3.17	S	1				1						
2.3.18	S	1	1						1	C2		
2.3.18.1	T											
2.3.19	Sub-S2	2	1	1					2	C2 C5		
2.3.20	Co-S	6		6					4	C7 C8(2) C9		
2.3.21	S	1	1						1	C5		

2.3.22	Sub-S2	2		1		1					1	
2.3.23	Sub-S2	3	1	1		1			1	C4		
2.3.24	Co-S	3			3				1	C8		
2.3.25	Sub-S5	2	1			1			1	C4		
2.3.26	Sub-S2	2		1	1							
2.3.27	Sub-S5	2	1	1								
2.3.28	cS+-	2		2					1	C8		
2.4	T											
2.4.1	Co-S	2		1	1				9	C6 C7 C8	1	
2.4.2	cS-+	2	1				1		1	C6		
2.4.3	Sub-S1	2	1	1								
2.4.4	S	1		1							1	
2.4.5a	T:											
2.4.5b	Sub-S1	2	1	1					1	C3	2	
2.4.6	S	1		1								
2.4.7a	T											
2.4.7b	Sub-S4	3		1		1	1		1	C8		
2.4.8	Sub-S1	2		1	1							
2.4.9	Sub-S1	3	1	2			3				1	
2.4.10	S	1		1								
2.4.11a	T											
2.4.11b	cS-+	2	2									
2.4.12	Sub-S3	2		2								
2.4.13a	T:										1	
2.4.13b	Sub-S1	2	1	1							2	
2.4.14	S	1	1								1	
2.4.15	Sub-S1	2	1	1					1	C3	1	
2.4.16	cS-+	2		1			1				1	
2.4.17	S	1			1				1	C9		
2.4.17.1	T											
2.4.17.2	T											
2.4.18	cS+-	2			2						2	
2.4.19	S	1	1									
2.4.20	S	1	1									
2.4.21	S	1		1								
2.4.22	S	1	1									
2.4.23	cS+-	2	1	1					1	C8		
2.4.23.1	S											
2.4.24a	cS	4	3	1					1	C8		
2.4.24b	Sub-S2											
2.4.25	cS-+	4	1	1			2					
2.4.26	Co-S	2	2						1	C8		
2.4.27	S	1		1								
2.4.28	Sub-S4	2				1	1					
2.4.29	S	1	1									
2.4.30	S	1		1								
2.4.31	S	1		1								
2.4.32	S	1		1					1	C7		
2.4.33	Sub-S2	2		1	1							
2.4.33.1	T											
2.4.34.	cS+-	2			2				1	C8	2	
2.4.35	S	1	1								1	
2.4.36	S	1		1								
2.4.37	S	1		1								

2.4.38	S	1		1								
2.4.39	S	1		1								
2.4.40	Sub-S1	2		1				1				
2.4.41	Sub-S1	2	2									
2.4.42	S	1				1					1	
2.4.43	S	1	1						1	C1		
2.4.44	S	1		1							1	
2.4.45	S	1		1								
2.4.46	S	1	1								1	
2.4.47	S	1	1								1	
2.4.48	Sub-S2	2		1	1				1	C1		
2.4.49	S	2	1	1					1	C9	1	
2.4.50	S	1			1						1	
2.4.51	Sub-S3	2		2								
2.4.51.1	T											
2.4.52	cS+-	2	1	1					1	C2		
2.4.53	cS+-	3	1				2		1	C6	1	
2.4.54	S	1		1					2	C6 C7		
2.4.55	Sub-S2	2		1	1							
2.5	T											
2.5.1	S	1	1									
2.5.2	S	1	1						1	C7	1	
2.5.3	S	1		1								
2.5.4	Sub-S3	2	1	1							1	
2.5.5	Sub-S1	4	2	2								
2.5.6	S	1		1								
2.5.7	Co-S	2	1		1							
2.5.8	Sub-S1	2	1	1							1	
2.5.9	S	1	1									
2.5.10	Sub-S1	2	1					1				
2.5.11	cS-+	2	1		1							

APPENDIX C3: Subordinate structures

Summary of occurrences

Text sections	N (Sub-s)
1	–
2.1	1
2.2	5
2.3	10
2.4	16
2.5	4
Total	36

Distribution of occurrences

Type	Occurrences	Total
Sub-S1	2.1.3 2.2.1a 2.3.3 2.3.6 2.4.3 2.4.5b 2.4.8 2.4.9 2.4.13b 2.4.15 2.4.40 2.4.48 2.5.8 2.5.10	14
Sub-S2	2.2.5 2.3.14 2.3.19 2.3.22 2.3.26 2.4.24b 2.4.33 2.4.48 2.4.55	9
Sub-S3	2.4.12 2.4.51 2.5.4	3
Sub-S4	2.2.7 2.4.7b 2.4.28	3
Sub-S5	a: 2.2.3 2.2.4 b: 2.3.25 c: 2.3.12 2.3.27 d: 2.5.5 e: 2.3.23	7
Total		36

APPENDIX C4: Nominal expressions

Summary of occurrences

Text sections	N (Ne)
1	3
2.1	28
2.2	6
2.3	3
2.4	9
2.5	1
Total	50

Distribution of occurrences

Type	Occurrences	Total
T	1 1.1 1.2 2 2.1 2.2 2.3 2.3.7.1 2.3.18.1 2.4 2.4.11a 2.4.17.1 2.4.17.2 2.4.33.1 2.4.51.1 2.5	16
T:	2.1.1.1 2.1.4a 2.1.5a 2.1.5b 2.1.6a 2.1.9.1 2.1.15.1 2.1.23.1 2.1.29 2.2.11a 2.2.12a 2.4.5a 2.4.13a	13
St	2.1.4b 2.1.5c 2.1.6b 2.1.11 2.1.12 2.1.13 2.1.17 2.1.19 2.1.25 2.1.27 2.1.28 2.1.31 2.2.9.1 2.2.9.2 2.2.9.3 2.4.24a	16
Sub-St	2.1.14 2.1.18 2.1.20 2.1.21 2.1.26	5
Total		50

APPENDIX C5: EMBEDDINGS

Summary of occurrences

Text sections	N (E)
1	–
2.1	1
2.2	6
2.3	4
2.4	5
2.5	1
Total	17

Distribution of occurrences

Type	Occurrences	Total
that S	2.1.9 2.2.1b 2.4.11b 2.5.11	4
[which] S	2.2.11b 2.2.13 2.4.14 2.3.8	4
[wh...]S rel.	2.2.12b 2.3.12	2
infinit.	2.2.5 2.2.14 2.3.14 2.4.2 2.4.9 2.4.16 2.4.25	7
Total		17

APPENDIX C6: Complex sentences

Summary of occurrences

Text sections	N (cS)
1	–
2.1	1
2.2	6
2.3	2
2.4	9
2.5	1
Total	19

Distribution of occurrences

Type	Occurrences	Total
cS + -	2.2.8 2.3.28 2.4.18 2.4.23 2.4.34 2.4.52 2.4.53	7
cS - +	2.1.9 2.2.1b 2.2.11b 2.2.12b 2.2.13 2.2.14 2.3.8 2.4.2 2.4.11b 2.4.16 2.4.25 2.5.11	12
Total		19

APPENDIX C7: COORDINATION

Summary of occurrences

	N	C1	C2	C3	C4	C5	C6	C7	C8	C9
1	10	1	1	2	1	–	–	2	3	–
2.1	18	4	-	4	–	3	–	4	2	1
2.2	5	–	2	1	–	–	–	–	2	–
2.3	20	–	4	–	5	2	1	1	4	3
2.4	20	2	1	2	–	–	4	3	6	2
2.5	1	–	–	–	–	–	–	1	–	–
Total	74	7	8	9	6	5	5	11	17	6

Distribution of occurrences

Type	Occurrences	Total
C1	1.1 2.1.12 2.1.15 2.1.17 2.1.29 2.4.43 2.4.48	7
C2	1.1.1 2.2.1a 2.2.8 2.3.8 2.3.16 2.3.18 2.3.19 2.4.52	8
C3	1.1.4 1.2.2 2.1.1 2.1.16 2.1.19 2.1.30 2.2.12b 2.4.5b 2.4.15	9
C4	1.1.3 2.3.10 2.3.13 2.3.14 2.3.23 2.3.25	6
C5	2.1.3 2.1.20 2.1.23 2.3.19 2.3.21	5
C6	2.3.1 2.4.1 2.4.2 2.4.53 2.4.54	5
C7	1.1.2 1.2.3 2.1.5b 2.1.28 2.1.29 2.1.31 2.3.2 2.4.1 2.4.32 2.4.54 2.5.2	11
C8	1.1.1(2) 1.1.2 2.1.1 2.1.21 2.2.4 2.2.5 2.3.20(2) 2.3.24 2.3.28 2.4.1 2.4.7b 2.4.23 2.4.24 2.4.26 2.4.34	17
C9	2.1.20 2.3.3 2.3.4 2.3.20 2.4.17 2.4.49	6
Total		74

Appendix D

APPENDIX D1: ANAPHORA

When necessary, we indicate the grammatical function of the anaphora (**subj**(ect) e.g. *this is* ... vs. **mod**(ifier) e.g. *this file* ...)

T UNIT	ANAPHORA FORM	COREFERENCE WITH	POSSIBILITY OF AMBIGUITY
1.1.1	that	NP	-
1.1.2	it	NP	-
1.2.1	this (subj)	NP	-
2.1.1	they	NP	-
2.1.3	it	NP	-
2.1.18	these (mod)	NP	-
2.2.3	its	NP	-
2.2.4	its	NP	-
2.2.6	this (mod)	several clauses	+
2.2.8	this (subj)	NP	-
2.3.5	this (subj)	clause	-
2.3.7	this (mod)	clause	-
2.3.11	this (subj)	NP	-
2.3.14	this (subj)	clause	-
2.3.15b	this (subj) its	clause NP	- -
2.3.20	their	NP	-
2.3.22	this (subj) it	clause NP	- +
2.3.23	it its	clause NP	- -
2.4.6	this (subj)	clause	-
2.4.11b	this (subj)	clause	-
2.4.14	this (subj)	clause	-

2.4.16	this (mod)	clause	-
2.4.24	this (subj)	clause	-
2.4.18	this (mod)	NP	-
2.4.33	these (mod)	several clauses	-
2.4.34	its	NP	-
2.4.48	these (mod)	several clauses	-
2.4.51	this (subj)	clause	-
2.5.7	it	NP	-
2.5.11	this	clause	-

APPENDIX D2: NEGATION

T UNIT	NEGATIVE FORM	NEGATED CONSTITUENT
2.1.4b	1	NP
2.1.5c	1	NP
2.1.6b	1	NP
2.1.18	1	V
2.1.20	2	
2.2.1a	1	V[pas]
2.2.1b	1	Adj
2.2.4	1	NP
2.2.6	2	
2.3.15b	1	V
2.3.19	1 1	Adj V[pas]
2.3.23	2	
2.4.45	1	V[pas]
2.4.48	1 1	V[pas] V
2.4.51	1	NP
2.4.55	1	V[pas]

APPENDIX E

APPENDIX E1: List of main verbs

The main verbs of the analyzed document (sections 1 and 2) are :

monitor	[3rd p.sg]
control	[3rd p.sg], [passive]
connect	[passive], [base], [infinitive]
produce	[base]
be	[3rd p.sg / pl], [infinitive], [base]
include	[3rd p.sg]
show	[passive]
detect	[passive]
contain	[3rd p.sg]
explain	[passive]
have	[3rd p.sg]
select	[passive], [3rd p.pl], [infinitive]
override	[3rd p.sg]
concern	[3rd p.pl]
require	[3rd p.sg]
perform	[base], [passive]
start	[passive],[base],[3rd p.sg / pl],[infinitive],[-ed participle]
command	[passive]
describe	[-ed participle], [passive]
base	[-ed participle]
do	[3rd p.sg], [passive]
want	[passive], [3rd p.sg]
stop	[infinitive], [passive], [base], 3rd p.sg / pl]
change	[passive], [-ing participle]
switch	[-ing participle], [passive], [3rd p.sg / pl]
enable	[passive]
mean	[3rd p.sg]
block	[-ed participle], [passive]
take place	[base]
run	[progressive]
reach	[perfective]
fail	[3rd p.sg]
decide	[infinitive]
use	[passive
suppose	[pseudopassive]
name	[-ed participle]
mount	[-ed participle]
call	[passive]
say	[passive]
occur	[3rd p.sg]
indicate	[passive]
synchronize	[passive], [infinitive], [3rd p.sg]
deload	[passive], [3rd p.sg]
get	[3rd p.sg]
update	[base]
depends on	[3rd p.sg]
adjust	[-ing participle]
close	[passive]
open	[base], [passive]
disconnect	[base], [passive]
take	[3rd p.sg]

satisfy	[passive]
lock	[passive]
deexit	[passive]
exit	[passive]
limit	[passive]
ignore	[base]
operate	[3rd p.sg]
know	[passive]
activate	[passive]
continue	[3rd p.sg]
release	[passive]
maintain	[base]
treat	[3rd p.sg]
remove	[passive]
measure	[3rd p.sg]
reservate	[3rd p.sg]

APPENDIX E2: LEXICAL ENTRIES

CONNECTION'

Occ	GramCat	Agent	Patient	Goal	
1.1.2	V[Pas]	**unknown**	NP[nom] *SG* **SG'**	PP[to] *ME* **ME'**	
1.1.4	N[n]	**PMS'**	PP[of] *SG* **SG'**	PP[to] *BT/ST* **BT' ∪ ST'**	
2.2.6	N[n]		PP[of] *MB* **MB'**	PP[to] *BB* **BB'**	interpretation discarded thanks to the lexicon
	N[n]	**PMS'**	**DG'**	**BB'**	
2.4.1	V[Base]	**PMS'∨ OP'**	NP[nom] *SG* **SG'**	PP[to] *BB or BT/ST* **BB' ∨ BT'∪ ST'**	
2.4.2	V[Inf]	**PMS'∨ OP'**	NP[obj] *SG* **SG'**	PP[to] *thrusters and BB* ***(BT'∪ ST') & BB**	
2.4.3	V[Pas]	**PMS'∨ OP'**	NP[nom] *SG* **SG'**	PP[to] *BB* **BB'**	
2.4.18	V[Base]	NP[nom] *operator* **OP'**	NP[obj] *SG* **SG'**	PP[to] *BB* **BB'**	
2.4.21	V[Pas]	**OP'**	NP[nom] *SG* **SG'**	PP[to] *BB* **BB'**	
2.4.28	V[Inf]	**OP'**	NP[obj] *SG* **SG'**	PP[to] *BB* **BB'**	

2.4.31	V[Pas]	**PMS'∨ OP'**	NP[nom] *DGs* **DGs'**	PP[to] *BB* **BB'**	
2.4.32	V[Pas]	**PMS'∨ OP'**	NP[nom] *SG* **SG'**	PP[to] *BB or ST* **BB'∪ ST'**	
2.4.33	V[Inf]	NP[nom] *PMS* **PMS'**	NP[obj] *SG* **SG'**	PP[to] *BB* **BB'**	
2.4.33'	V[Pas]	**OP'**	NP[nom] *SG* **SG'**	PP[to] *thrusters* **BT'∪ ST'**	
2.4.34	V[Base]	*operator* **OP'**	NP[obj] *SG* **SG'**	PP[to] *BT and/or ST* **BT'∪ ST'**	
2.4.35	N[n]	**OP'**	**SG'**	PP[to] *BT and/or ST* **BT'∪ ST'**	
2.4.42	N[n]		PP[of] *BT and/or ST*		discarded interpretation
	N[n]	**OP'**	**SG'**	PP[to] *BT and/or ST* **BT'∪ ST'**	
2.4.45	V[Pas]	**PMS'∨ OP'**	NP[nom] *SG* **SG'**	PP[to] *BB* **BB'**	
2.4.48	V[Base]	NP[nom] *PMS* **PMS'∨ OP'**	NP[obj] *SG* **SG'**	PP[to] *thrusters* **BT'∪ ST'**	
2.4.53	V[Inf]	PP[via] *ISC console* **OP'**	NP[obj] *SG* **SG'**	PP[to] *... BB ...* *thrusters* **BB' ∨** **(BT'∪ ST')**	
2.4.54	N[n]	**PMS'**	**SG'**	PP[to] *BB and thrusters* **BB' ∨** **(BT'∪ ST')**	

2.4.55	N[n]	**PMS'**	**SG'**	**BB' ∨ (BT'∪ ST')**	

CONNECTION

- connection; N[n = nominalization]
- **connection'**(ag(x), pat(y), goal(z))
- ((<PP[by], [+Hum or +Soft]>$_1$), (<PP[of], [-Anim]>$_2$), (<PP[to], [-Anim]>$_3$))
- <u>Reading</u> C_1 == ag(x)
  C_2 == pat(y)
  C_3 == goal(z)

connection'(ag(x), pat(y), goal(z))

(C_1) (C_2) (C_3)

E.g. *the connection (of the magnetoscope) (to the TV) (by [Peter, the program])*

CONNECT

- connect; V[act]
- **connection'**(ag(x), pat(y), goal(z))
- (a) (<NP[nom], [+Hum or +Soft]>$_1$, <NP[obj], [-Anim]>$_2$, (<PP[to], [-Anim]>$_3$))
- (a) <u>Reading</u> C_1 == ag(x)
  C_2 == pat(y)
  C_3 == goal(z)

connection'(ag(x), pat(y), goal(z))

C_1 C_2 (C_3)

E.g. *Peter connects the TV (to the magnetoscope)*

- (b) (<NP[nom],[-Anim]>$_1$, <PP[to], [-Anim]>$_2$)
- (b) <u>Reading</u> C_1 == pat(x)
  C_2 == goal(y)

connection'(ag(x), pat(y), goal(z))

Ø C_1 C_2

E.g. *The TV connects to the magnetoscope*

- **connection-phys'**(ag(x), pat(y), goal(z))

x' = unknown
y' = SG'
z' = ME'

• only used in agentless passives

• in SRS of the type A –> B **connection-phys'** is never used in the expression of A (e.g. **when the SG is connected to the ME*).

- **connection-fl'**(ag(x), pat(y), goal(z))

$x' \in \{PMS', OP'\}$
$y' \in \{SG', DG1', DG2', DG3'\}$
$z' \in \{BB', BT', ST'\}$

$(y' = SG') \rightarrow (z' = (BB' \vee (BT' \gg ST')))$
$(y' = DGi') \rightarrow z' = BB'$

-

Denotations	Linguistic expressions
$BT' \cup ST'$	BT/ST BT or ST BT and/or ST thrusters
$BB' \vee (BT' \cup ST')$	BB or BT/ST BB and thrusters
OP'	operator ISC console

SYNCHRONIZATION'

Occ	GramCat	Agent	Patient	Goal	
1.13	N[n]	**PMS'**	**DG'**		
1.1.4	N[n]	**PMS'**	**SG'**	PP[to] *BB* **BB'**	
2.1.14	N[n]	**PMS'**	**DG'**		
2.1.25	N[n]	**PMS'**	PP[of] *SG* **SG'**	PP[to] *BB* **BB'**	
2.2.6	N[n]	? PP[by] *the PMS system* **PMS'**	PP[of] *MB* **(DG's)MB'**	PP[to] *BB* **BB'**	
2.3.4	V[Prp]	**MSB'**	PP[of] *MB* **(DG's)MB'**		
2.3.18	N[n]	**OP'**	**DG'**		
2.3.20	V[Pas]	**PMS'**	NP[nom] *DGs* **DGs'**		
2.4.5	N[n]	**MSB'**	PP[of] *MB* **(SG's)MB'**	**BB'**	
2.4.7 (a)	V[Inf]	**MSB'**	NP[nom] *DGs* **DGs'**		discarded interpretation
	V[Inf]	**MSB'**	**SG'**		
2.4.7 (b)	N[n]	**MSB'**	**BB' & DGs'**	**SG'**	
2.4.10	N[n]	**PMS'**	**BB' & DGs'**	**SG'**	

2.4.20	V[Fin]	NP[nom] *frequency controlled DG*	NP[obj] *BB*	PP[to] *SG*	interpretation induced via the lexicon
		OP'	**fr.c.DG' & BB'**	**SG'**	

SYNCHRONIZATION

• synchronization; N[n]

• **synchronization'**(ag(x), pat(y), goal(z))

• ((<PP[by], [+Hum or +Soft]>$_1$), (<PP[of], [-Anim]>$_2$), (<PP[to], [-Anim]>$_3$))

• Reading C_1 == ag(x)
C_2 == pat(y)
C_3 == goal(z)

synchronization'(ag(x), pat(y), goal(z))

(C_1) (C_2) (C_3)

E.g. *the synchronization (of the generator) (to the others) (by [Peter, the program])*

SYNCHRONIZE

• synchronize; V[act]

• **synchronization'**(ag(x), pat(y), goal(z))

• (a) (<NP[nom], [+Hum or +Soft]>$_1$, <NP[obj], [-Anim]>$_2$, (<PP[to], [-Anim]>$_3$))

• (a) Reading C_1 == ag(x)
C_2 == pat(y)
C_3 == goal(z)

synchronization'(ag(x), pat(y), goal(z))

C_1 C_2 (C_3)

E.g. *Peter synchronizes the generator (to the others)*

• (b) (<NP[nom],[-Anim]>$_1$, (<PP[to], [-Anim]>$_2$))

• (b) Reading C_1 == pat(x)
C_2 == goal(y)

synchronization'(ag(x), pat(y), goal(z))

$$\begin{array}{ccc} \mathrm{ag}(x) & \mathrm{pat}(y) & \mathrm{goal}(z) \\ | & | & | \\ \emptyset & C_1 & (C_2) \end{array}$$

E.g. *The generator synchronizes (to the others)*

- **synchronization'**(ag(x), pat(y), goal(z))

x' ∈ {PMS', OP', MSB'}
y' ∈ {SG', DG1', DG2', DG3', BB'}
z' ∈ {BB', SG'}

y' = SG' -> z' = BB'
y' = DGi' -> z' = BB'
y' = DGi' & BB' -> z' = SG'

• mainly used in nominalized forms

• subcategorization for **PP[to]** instead of **PP[with]** because the PATIENT's frequency is the one which is changing whereas the GOAL's frequency is the reference frequency

Denotations	Linguistic expressions
SG'	SG MB
DGi'	DGs MB frequency controlled DG
PMS'	PMS the PMS system

CONTROL'

Occ	GramCat	Agent	Patient	
1.1.1	V[Fin]	NP[nom] *PMS* **PMS'**	NP[obj] *electricity production* **electricity_production'**	

1.1.3	N[n]		NP control *frequency control* **(DG's) frequency'**	
1.2.1	V[Psp]/ADJ		controlled NP *controlled system /components* **system' /components'**	
2.1.0 (title)	N[n]	**PMS'v OP' v MSB'**	**SG'v DGs'**	control NP *control modes*
2.1.4	N[n]	***PMS'v OP'**	PP[of] *DG* **DG'**	MANUAL
2.1.5	N[n]	***PMS'v OP'**	PP[of] *thruster MB* **(SG's)MB' to BT/ST'**	MANUAL
2.1.6	N[n]	***PMS'v OP'**	PP[of] *MB* **(SG's)MB' to BB'**	MANUAL
2.1.12	N[n]	**PMS'**	PP[of] *online DGs* **online DGs' frequency'**	NP control *frequency control* DG SEMIAUTOMATIC DG AUTOMATIC
2.1.31	N[n]	**PMS'**	PP[of] *SGs* **SG'**	SG AUTOMATIC
2.2.10	N[n]	**PMS'v OP'**	**DGs'**	control NP *PMS control modes*
2.2.11 2.2.13	V[Pas]	**PMS'**	NP[nom] *online DG* **1 online DG's frequency'**	NP controlled *frequency controlled* DG SEMIAUTOMATIC DG AUTOMATIC

2.3.1	V[Pas]	PP[on] *the AE,* PP[from] *the MSB* or PP[from] *the PMS* **AE'∨ MSB' ∨ PMS'**	NP[nom] *DGs* **DGs'**	
2.3.2	N[n]	**AE'∨ MSB' ∨ PMS'**	**DGs'**	control NP *control possibilities*
2.3.4	V[Pas]	PP[from] *the MSB* or PP[on] *the AE* **MSB'∨ AE'**	NP[nom] *the DG* **DG'**	DG MANUAL
2.3.5	N[n]	**MSB'**	NP control *speed/load control* **DGs'**	DG MANUAL
2.3.6	N[n]	NP control *PMS control* **PMS'**	**DG'**	DG AUTO
2.3.7	N[n]	**ISC-system'**	**DG'**	DG AUTO
2.3.8	V[Fin]	NP[nom] *the operator* PP[from] *the ISC consoles* **OP'**	NP[obj] *which DGs are online and stopped* **online ∨ stopped DGs'**	DG SEMIAUTOMATIC
2.3.9	V[Fin]	NP[nom] *the operator* **OP'**	NP[obj] *the available power* **available power'**	DG SEMIAUTOMATIC
2.3.10 (a)	V[Psp]/ADJ	NP controlled NP *PMS controlled ...* **PMS'**	NP controlled NP *... controlled DG* **DG'**	DG SEMIAUTOMATIC
2.3.10 (b)	V[Pas]	**PMS'**	NP[nom] *PMS controlled DG* **1 online DG with highest priority frequency'**	NP controlled *frequency controlled* DG SEMIAUTOMATIC
2.3.13	V[Psp]/ADJ	NP controlled NP *PMS controlled ...* **PMS'**	NP controlled NP *... controlled DGs* **DGs'**	DG SEMIAUTOMATIC

2.3.14	V[Psp]/ADJ	NP controlled NP *PMS controlled ...* **PMS'**	NP controlled NP *... controlled DG* **DG'**	DG SEMIAUTOMATIC
2.3.19	N[n]	NP control *PMS control* **PMS'**	**DG'**	DG AUTOMATIC
2.3.21 2.3.22 2.3.23 2.3.25	V[Psp]/ADJ	NP controlled NP *PMS controlled ...* **PMS'**	NP controlled NP *...controlled DG(s)* **DG(s)'**	DG AUTOMATIC
2.3.26	N[n]	NP control *PMS control* **PMS'**	**DGs'**	NP control NP *PMS control modes*
2.4.1	N[n]	**PMS'∨ MSB'**	**SG'**	
2.4.4	N[n]	**PMS'∨ MSB'**	**SG'**	control NP *control possibilities*
2.4.9	N[n]	NP control *PMS control* **PMS'**	**SG MB'**	SG AUTO
2.4.11	N[n]	NP control *PMS control* **PMS'**	**DG online'**	DG AUTO
2.4.13	V[Pas]	PP[from] *the MSB* **MSB'**	NP[nom] *the MB* **SG MB' to BT/ST'**	SG MANUAL
2.4.14	N[n]	**MSB'**	PP[of] *SG voltage* **SG voltage'**	SG MANUAL
2.4.15	V[Pas]	PP[by] *the PMS* **PMS'**	NP[nom] *the MB and the power up procedure for ST/BT* **SG MB' & power up for ST/BT'**	SG AUTO
2.4.20	V[Psp]/ADJ		NP controlled NP *frequency controlled DG* **DG's frequency'**	SG SEMIAUTOMATIC

2.4.52	V[Pas]	PP[from] *the ship handling mode selector* **ship handling mode selector'**	NP[nom] *the SG operation* **SG operation'**	SG AUTOMATIC
2.5.1	V[Fin]	NP[nom] *the PMS system* **PMS'**	NP[obj] *power reservation* **power reservation'**	

CONTROL

• control; N[n]

• **control'**(ag(x), pat(y), instr(z))

• (a) ((<NP[nom], [+Hum or +Soft]>$_1$), (<PP[of], [-Anim]>$_2$))

• (a) <u>Reading</u> $C_1 == ag(x)$
$C_2 == pat(y)$

control'(ag(x), pat(y), instr(z))

(C_1) (C_2) Ø

E.g. *the (PMS) control (of electricity production)*

• (b) ((<NP[nom], [-Anim]>$_1$), (<PP[of], [-Anim]>$_2$))

• (b) <u>Reading</u> $C_1 == pat(y)$
$C_2 == pat(y)$

control'(ag(x), pat(y), instr(z))

Ø ($C_{1,2}$) Ø

E.g. *the ([frequency, speed/load]) control (of online DGs)*

• (c) ((<NP[nom], [+Hum or +Soft]>$_1$), <NP[nom], [-Anim]>$_2$)

• (c) <u>Reading</u> $C_1 == ag(x)$
$C_2 == instr(z)$

control'(ag(x), pat(y), instr(z))

(C_1) Ø C_2

E.g. *the (PMS) control modes*

CONTROL

- control; V[act]
- **control'**(ag(x), pat(y), instr(z))
- (<NP[nom], [+Hum or +Soft]>$_1$, <NP[obj], [-Anim]>$_2$, (<PP[in], [-Anim]>$_3$))
- <u>Reading</u> C_1 == ag(x)
  C_2 == pat(y)
  C_3 == instr(z)

control'(ag(x), pat(y), instr(z))

C_1 C_2 (C_3)

E.g. *(In AUTOMATIC mode), the PMS controls the SG's MB*

CONTROLLED

- control; V[Pas]
- **control'**(ag(x), pat(y), instr(z))
- (<NP[nom], [-Anim]>$_1$, (<ADV, [-Anim]>$_2$), (<PP[by,on,from], [+Hum or +Soft]>$_3$), (<PP[in], [-Anim]>$_4$))
- <u>Reading</u> C_1 == pat(y)
  C_2 == pat(y)
  C_3 == ag(x)
  C_4 == instr(z)

control'(ag(x), pat(y), instr(z))

(C_3) C_1,(C_2) (C_4)

E.g. *the online DG is (frequency) controlled (by the PMS) (in AUTOMATIC mode)*

CONTROLLED

• control; ADJ

• **control'**(ag(x), pat(y), instr(z))

• ((<NP[nom], [+Hum or +Soft]>$_1$), <NP[nom], [-Anim]>$_2$, (<ADV, [-Anim]>$_3$))

• <u>Reading</u> C_1 == ag(x)
C_2 == pat(y)
C_3 == pat(y)

control'(ag(x), pat(y), instr(z))

(C_1) C_2,(C_3) Ø

E.g. *the (PMS) (frequency) controlled DG*

• **control'**(ag(x), pat(y), instr(z))

x' ∈ {PMS', OP', MSB', AE', ship handling mode selector'}

The relations between the different elements of this set remain unclear and have to be checked with an expert.

y' ∈ {power production', component'}

z' ∈ {selector', mode'}

z' = MANUAL position' ↔ z' = MANUAL mode'

z' = MANUAL position' & y' = DG' –> x' = MSB'∨ AE'

z' = MANUAL position' & y' = SG' –> x' = MSB'

z' = AUTO position' ↔ z' = AUTOMATIC mode' ∨ SEMIAUTOMATIC mode'

z' = AUTOMATIC mode' –> x' = ship handling mode selector' & PMS'

z' = SEMIAUTOMATIC mode' –> x' = PMS' & OP'

• mainly used in nominalized forms

• lots of passives

• special construction (cf. CONTROL (b), CONTROLLED (V) and CONTROLLED (ADJ)) where some kind of 'internal' object (i.e. the physical patient of control) is mentioned, whereas in general it is the component (e.g. DG or SG) which is said to be the patient : some sort of metonymy ? There is thus a specific technical definition of the notion of *frequency controlled DGi*, i.e. the DG whose frequency is the reference frequency for the others DGs : "The online, PMS controlled DG with the highest priority is frequency controlled" (2.3.10)

• <u>Denotations</u> <u>Linguistic expressions</u>

PMS'	PMS PMS system
OP'	the operator the operator from the ISC consoles the ISC-system
power production'	electricity production frequency speed/load online and stopped the available power voltage power up procedure for ST/BT power reservation
component'	system DG thruster MB MB online DG(s) SG (PMS) controlled DG the SG operation
selector'	MANUAL/AUTO selector MANUAL/AUTO switch MANUAL position AUTO position AUTO mode (highly confusing, cf. below)
mode'	MANUAL mode AUTOMATIC mode SEMIAUTOMATIC mode

START' vs. STOP'

Occ	GramCat	Agent	Patient	
1.1.3 (a)	N[n]	PMS'v OP'	DG'	*start/stop system*
1.1.3 (b)	N[n]	PMS'	DG'	*black out start system*
2.1.11	N[n]	PMS'	DG'	*black out start*
2.1.13 (a)	N[n]	PMS'	DG'	*start attempt*
2.1.13 (b)	N[ing]	PMS'	DG'	*starting failure*
2.1.14	V[Pas]	PMS'	NP[nom] *the diesel engine* DG'	*started*
2.1.15 (a)	N[n]	OP'	PP[of] *DGs* DGs'	*start and stop*
2.1.15 (b)	N[n]	PMS'	DGs'	*black out start*
2.1.17	N[n]	PMS'	PP[of] *DGs* DGs'	*start and stop*
2.1.19	N[n]	PMS'	PP[of] *stanby DG* DG'	*start*
2.1.20 (a)	N[n]	PMS'	PP[of] *one or twoDGs* DG(s)'	*start*
2.1.20 (b)	V[Pas]	PMS'	NP[nom] *SG* SG'	*stopped*
2.1.21 (a)	N[n]	PMS'	PP[of] *two DGs* DGs'	*start*

2.1.21 (b)	N[n]		**SG'**	*stanby start shutdown*
2.1.26	N[n]	**PMS'**	PP[of] *DGs online* **DGs'**	*stop*
2.1.27	N[n]	**PMS'**	**SG' (BT/ST online')**	*start sequence*
2.1.28	N[n]	**PMS'**	**SG'**	*stop sequences*
2.1.29	N[n}	**OP'**	PP[of] *SG* **SG'**	*start and stop*
2.2.1	N[n]	**PMS'**	**DG'**	*blakout start*
2.2.4	V[Pas]	**PMS'**	NP[nom] *the first (DG)* **DG'**	*started*
2.2.5 (a)	V[Pas]	**PMS'**	NP[nom] *the next DG* **DG'**	*started*
2.2.5 (b)	V[Inf]	**PMS'**	NP[nom] *the former DG* **DG'**	*to start*
2.2.14	V[Pas]	**PMS'**	NP[nom] *which DG* **DG'**	*is started*
2.3.9	V[Pas]	**OP'**	NP[nom] *which DGs* **DGs'**	*stopped*
2.3.14	V[Inf]	NP[nom] *the operator* **OP'**	NP[obj] *DG* **DG'**	*to stop*
2.3.16 (a)	N[ing]	**OP'**	**DG'**	*stopping*

2.3.16 (b)	N[ing	**OP'**	PP[of] *engine* **DG'**	*stopping*
2.3.17 (a)	N[n]	**ISC console'**	PP[of] *DG* **DG'**	*start*
2.3.17 (b)	V[Psp]/ADJ	**ISC console'**	NP[nom] *DG* **DG'**	*stopped*
2.3.18 (a)	N[ing]	**ISC console'**	**DG'**	*starting*
2.3.18 (b)	N[ing]	**ISC console'**	PP[of] *engine* **DG'**	*starting*
2.3.20	V[Pas]	**PMS'**	NP[nom] *the following DGs* **DGs'**	*started and stopped*
2.3.22	N[n]	**PMS'**	**DG'**	*start/stop sequence*
2.3.23	N[n]	**PMS'**	**DG'**	*stop*
2.3.24 (a)	V[Base]	NP[nom] *the PMS* **PMS'**	NP[obj] *a new DG* **DG'**	*start*
2.3.24 (b)	V[Base]	NP[nom] *the PMS* **PMS'**	NP[obj] *the one in question* **DG'**	*stop*
2.3.25 (a)	N[n]	**PMS'**	PP[of] *DG* **DG'**	*start*
2.3.25 (b) and (c)	V[Psp]/ADJ	**PMS'**	NP[nom] *DG* **DG'**	*stopped*
2.3.27 (a)	N[n]	**PMS'**	**DG'**	*standby start*
2.3.27 (b)	V[Pas]	**PMS'**	NP[nom] *standby DG* **DG'**	*started*

2.3.28	V[Pas]	**PMS'**	NP[nom] *DG* **DG'**	*started*
2.4.8	V[Bse]	NP[nom] *the operator* **OP'**	NP[obj] *DGs* **DGs'**	*stop*
2.4.12	V[Pas]	**PMS'∨ OP'**	NP[nom] *the DGs* **DGs'**	*stopped*
2.4.23	V[Fin]	**OP'**	NP[nom] *DGs* **DGs'**	*stops*
2.4.24	V[Fin]	**PMS'**	NP[nom] *one or two DG(s)* **DG(s)'**	*starts*
2.4.25 (a)	V[Inf]	NP[nom] *the operator* **OP'**	NP[obj] *the DGs* **DGs'**	*to start*
2.4.25 (b)	V[Inf]	**OP'**	NP[nom] *SG* **SG'**	*to stop*
2.5.2 (a)	N[n]	**PMS'**	PP[from] *power consumer* **power consumer'**	*start request*
2.5.2 (b)	N[n]	**PMS'**	PP[to] *power consumer* **power consumer'**	*start blocking*
2.5.4 2.5.5 (a)	N[n]	**PMS'**	**power consumer'**	*start blocking*
2.5.5 (b)	V[Pas]	**PMS'∨ OP'**	NP[nom] *standby DG* **DG'**	*is started*
2.5.6	N[n]	**PMS'**	PP[to] *power consumer* **power consumer'**	*start blocking*

2.5.7 (a)	V[Pas]	PMS'	NP[nom] *power consumer* **power consumer'**	*started*
2.5.7 (b) 2.5.8 (b)	N[n]	PMS'	**power consumer'**	*start request*
2.5.7 (a)	V[Fin]	PMS'	NP[nom] *consumer* **consumer'**	*stops*

START vs. STOP

• start vs. stop; N[n]

• **start'** vs. **stop'**(ag(x), pat(y))

• ((<PP[of], [-Anim]>$_1$))

• Reading C_1 == pat(y)

start' vs. **stop'**(ag(x), pat(y))

Ø (C1)

E.g. *the start vs. stop (of the engine)*

STARTING vs. STOPPING

• starting vs. stopping; N[n]

• **start'** vs. **stop'**(ag(x), pat(y))

• ((<PP[of], [-Anim]>$_1$))

• Reading C_1 == pat(y)

start' vs. **stop'**(ag(x), pat(y))

Ø (C1)

E.g. *the startingvs.stopping (of the engine)*

START vs. STOP

- start vs. stop; V[act]
- **start'** vs. **stop'**(ag(x), pat(y))
- (a) (<NP[nom], [+Hum or +Soft]>$_1$, <NP[obj], [-Anim]>$_2$)
- (a) <u>Reading</u> C_1 == ag(x)
C_2 == pat(y)

$$\textbf{start'}\ \text{vs.}\ \textbf{stop'}(\text{ag}(\underset{C_1}{x}),\ \text{pat}(\underset{C_2}{y}))$$

E.g. *the operator starts vs. stops the engine*

• (b) (<NP[nom], [+Hum or +Soft]>$_1$, <NP[obj], [-Anim]>$_2$)

• (b) <u>Reading</u> C_1 == pat(y)

$$\textbf{start'}\ \text{vs.}\ \textbf{stop'}(\text{ag}(\underset{\varnothing}{x}),\ \text{pat}(\underset{C_1}{y}))$$

E.g. *the engine starts vs. stops*

<u>STARTED vs. STOPPED</u>

• started vs. stopped; ADJ

• **start'** vs. **stop'**(ag(x), pat(y))

• (<NP[nom], [-Anim]>$_1$)

• <u>Reading</u> C_1 == pat(y)

$$\textbf{start'}\text{vs.}\textbf{stop'}(\text{ag}(\underset{\varnothing}{x}),\ \text{pat}(\underset{C_1}{y}))$$

E.g. *the started engine*

• **start'** vs. **stop'**(ag(x), pat(y))

x' $\in$ {PMS', OP'}
y' $\in$ {SG', DG1', DG2', DG3', power consumer'}

start'(ag(x1), pat(y1)) -> **stop'**(ag(x2), pat(y2))
where (y1' = DG1...n' & y2' = SG') or (y1' = SG' & y2' = DG1...n')

blackout start -> x' = PMS'

<u>Denotations</u>	<u>Linguistic expressions</u>
DGi'	DG the diesel engine standby DG one or two DGs two DGs DGs online the first DG the next DG the former DG engine the following DGs a new DG the one in question
power consumer'	power consumer consumer

SWITCH'

Occ	GramCat	Agent	Patient	Instrument	
1.2.3	N[n]				Emergency Switch Board EMS
2.1.17	V[ing]	**PMS'**	NP[obj] *BT/ST* **BT'/ST'**	ADV *online* **online'**	
2.1.17	V[ing]	**PMS'**	NP[obj] *SG* **SG'**	ADV *offline* **offline'**	
2.2.3	V[Pas]	**PMS'**	NP[nom] *the highest priorited (DG)* **master DG'**	ADV *online* **online'**	
2.2.5	V[Inf]	**PMS'**	NP[nom] *the former DG* **DG'**	ADV *online* **online'**	
2.2.6	N[n]	**PMS'**	**DG'**	ADV *online* **online'**	

2.3.2 (a)	N[ing]		**DG'**	PP[between] *different control possibilities* **MANUAL'/ AUTO'**	
2.3.2 (b)	N[n]	? **MSB'**	**DG'**	NP[nom] *MANUAL/ AUTO switch* **MANUAL'/ AUTO'**	MANUAL/ AUTO switch mounted in the MSB
2.3.3	N[n]		? **DG'**	PP[in] *MANUAL position* **MANUAL'**	
2.3.6	N[n]		? **DG'**	PP[in] *AUTO position* **AUTO'**	
2.3.16	N[ing]	**OP'**	PP[of] *engine* **DG'**	ADV *offline* **offline'**	
2.3.18	N[ing]	**OP'**	PP[of] *engine* **DG'**	ADV *online* **online'**	
2.3.20 (a)	V[Pas]	**PMS'**	NP[nom] *DGs* **DGs'**	ADV *online* **online'**	
2.3.20 (b)	V[Pas]	**PMS'**	NP[nom] *DGs* **DGs'**	ADV *offline* **offline'**	
2.3.22 (b)	V[ing]		NP[obj] *it* **DG'**	PP[to] *MANUAL mode* **MANUAL'**	
2.4.4 (a)	N[ing]		**SG'**	PP[between] *control possibilities* **MANUAL' /AUTO'**	

2.4.4 (b)	N[n]	? **MSB'**	**SG'**	**MANUAL'** **/AUTO'**	MANUAL/ AUTO switch mounted in the MSB
2.4.5	N[n]		? **SG'**	PP[in] *MANUAL position* **MANUAL'**	
2.4.7	VP[Pas]		NP[nom] *the DGs* **DGs'**	PP[to] *MANUAL* **MANUAL'**	
2.4.9 2.4.15	N[n]		? **SG'**	PP[in] *AUTO position* **AUTO'**	
2.4.13	N[n]		? **SG'**	PP[in] *MANUAl position* **MANUAL'**	
2.4.24	V[Fin]	**PMS'**∨ **OP'**	NP[nom] *one or two DGs* **DGs'**	ADV *online* **online'**	
2.4.37	V[Pas]	**OP'**	NP[nom] *SG* **SG'**	PP[to] *current mode* **current mode'**	
2.4.40	V[Pas]	**OP'**	NP[nom] *SG* **SG'**	PP[to] *voltage mode* **voltage mode'**	
2.5.5	V[Pas]	**PMS'**	NP[nom] *a standby DG* **DG'**	**ADV** *online* **online'**	

<u>SWITCH</u>

• switch; N[n]

• **switch'**(ag(x), pat(y), instr(z))

• (a) ((<ADV, [+Position]>$_1$))

• (a) <u>Reading</u> C_1 == instr(z)

$$\begin{array}{cccc} \textbf{switch'}(& ag(x), & pat(y), & instr(z)) \\ & | & | & | \\ & \varnothing & \varnothing & (C_1) \end{array}$$

E.g. *the switch (online)*

• (b) $((<PP[in], [+Position]>_1))$

• (b) <u>Reading</u> $C_1 == instr(z)$

$$\begin{array}{cccc} \textbf{switch'}(& ag(x), & pat(y), & instr(z)) \\ & | & | & | \\ & \varnothing & \varnothing & (C_1) \end{array}$$

E.g. *the switch (in MANUAL position)*

• (c) $((<NP[nom], [+Position]>_1))$

• (c) <u>Reading</u> $C_1 == instr(z)$

$$\begin{array}{cccc} \textbf{switch'}(& ag(x), & pat(y), & instr(z)) \\ & | & | & | \\ & \varnothing & \varnothing & (C_1) \end{array}$$

E.g. *the (MANUAL/AUTO) switch*

<u>SWITCHING</u>

• switching; N[ing]

• **switch'**(ag(x), pat(y), instr(z))

• (a) $(<PP[between], [+Position]>_1)$

• (a) <u>Reading</u> $C_1 == instr(z)$

$$\begin{array}{cccc} \textbf{switch'}(& ag(x), & pat(y), & instr(z)) \\ & | & | & | \\ & \varnothing & \varnothing & (C_1) \end{array}$$

E.g. *the switching between different possibilities*

• (b) $((<PP[of], [-Anim]>_1), <ADV, [+Position]>_2)$

• (b) <u>Reading</u> $C_1 == pat(y)$
$C_2 ==instr(z)$

switch'(ag(x), pat(y), instr(z))

Ø (C$_1$) C$_2$

E.g. *the switching online (of the engine)*

<u>SWITCH</u>

• switch; V[act]

• **switch'**(ag(x), pat(y), instr(z))

• (a) (<NP[obj], [-Anim]>$_1$, <ADV or PP[to], [-Anim]>$_2$, (<NP[nom], [+Hum or +Soft]>$_1$))

• (a) <u>Reading</u> C$_1$ == pat(y)
C$_2$ == instr(z)
C$_3$ == ag(x)

switch'(ag(x), pat(y), instr(z))

(C$_3$) C$_1$ C$_2$

E.g. ... *for switching SG online ...*

• (b) (<NP[nom], [-Anim]>$_1$, <ADV or PP[to], [-Anim]>$_2$)

• (b) <u>Reading</u> C$_1$ == pat(y)
C$_2$ == instr(z)

switch'(ag(x), pat(y), instr(z))

Ø C$_1$ C$_2$

E.g. *the DG switches online*

• **switch'**(ag(x), pat(y), instr(z))

x' $\in$ {PMS', OP', MSB', EMS'}
y' $\in$ {SG', BT'/ST', DG1', DG2', DG3'}
z' $\in$ {online'/offline', MANUAL'/AUTO', current'/voltage'}

z' = online'/offline' –> x' = PMS' $\vee$ OP'
z' = current'/voltage' –> y' = SG' & x' = OP'
z' = MANUAL'/AUTO' –> x' = MSB' (?)
x' = EMS' –> ???

• verb only used in agentless forms (i.e. V[Pas] or V[ing] for reading (a))

Denotations	Linguistic expressions
DGi'	DG(s) the highest priorited the first in the standby sequence the former DG engine it one or two DGs a standby DG
MANUAL'/AUTO'	different control possibilities MANUAL position AUTO position MANUAL mode control possibilities MANUAL
current'/voltage'	current mode voltage mode

SELECT'

Occ	GramCat	Agent	Patient	Instrument
2.1.2	N[n]		GS'	NP[n] *MANUAL/AUTO selector* **MANUAL' /AUTO'**
2.1.3	V[Pas]			NP[nom] *MANUAL mode* **MANUAL mode'**
2.1.3	N[n]	NP[nom] *ship handling mode selector* **ship handling mode selector'**		**with/without SG mode'**
2.1.23	V[Psp]/ADJ		SG'	NP[nom] *mode SG SEMIAUTOMATIC and SG AUTOMATIC* **(SEMI)AUTO MATIC'**

2.1.31 (a)	V[Psp]/ADJ	PP[on] *ship handling mode selector* **ship handling mode selector'**		NP[nom] *mode* **mode'**
2.1.31 (b)	N[n]	NP[nom] *ship handling mode selector* **ship handling mode selector'**		**ship handling mode'**
2.2.8	V[Pas]	PP[from] *the ISC console* **OP'**	NP[nom] *this* **DG priority'**	**priority 1/2/3'**
2.2.11	V[Inf]	PP[from] *the ISC console* **PMS'∨ OP'**	NP[obj] *online DG freq. controlled* **DG'**	**priority'**
2.2.12 2.2.14	V[Inf]	**PMS'**	NP[obj] *DG* **DG'**	**priority'**
2.3.19	V[Pas]		NP[nom] *SG operation to BB* **SG operation to BB'**	

<u>SELECTOR</u>

- selector; N[n]
- **select'**(ag(x), pat(y), instr(z))
- (<NP[nom], [+Position]>1)

• Reading $C_1 == instr(z)$

select'(ag(x), pat(y), instr(z))

Ø Ø (C_1)

E.g. *the (MANUAL/AUTO) selector*

SELECT

• select; V[act]

• **select'**(ag(x), pat(y), instr(z))

• (<NP[obj], [-Anim]>$_1$, (<NP[nom], [+Hum or +Soft]>$_2$))

• Reading $C_1 == pat(y)$
$C_2 == ag(x)$

select'(ag(x), pat(y), instr(z))

(C_2) C_1 Ø

E.g. ... *in order to select the online DG* ...

SELECTED

• selected; ADJ

• **select'**(ag(x), pat(y), instr(z))

• (<NP[nom], [+Position]>$_1$, (<PP[on], [+Hum or +Soft]>$_2$))

• Reading $C_1 == instr(z)$
$C_2 == ag(x)$

select'(ag(x), pat(y), instr(z))

(C_2) Ø C_1

E.g. *the selected mode AUTO (on ship handling mode selector)*

- **select'**(ag(x), pat(y), instr(z))

x' ∈ {PMS', OP', ship handling mode selector'}
y' ∈ {SG', DG1', DG2', DG3'}
z' ∈ {MANUAL'/AUTO', with/without SG', AUTOMATIC'/SEMIAUTOMATIC', priority 1/2/3'}

z' = priority 1/2/3'–> x' = PMS'∨ OP'
z' = with/without SG'–> x' = ship handling mode selector'

• verb only used in agentless forms (i.e. V[Pas] or V[Inf])

- <u>Denotations</u> — <u>Linguistic expressions</u>

Denotations	Linguistic expressions
DGi'	DG online frequency controlled DG
SG'	SG operation to BB

(DE)LOAD'

Occ	GramCat	Agent	Patient	
1.1.3	N[n]		**DGs'**	*loadsharing system*
2.1.12	N[n]	**PMS'**	PP[of] online DGs **DGs'**	*loadsharing*
2.1.20	N[n]		**one or two DGs'**	*load dependent*
2.3.4	N[n]		**DG'**	*load control*
2.3.13	N[n]		**DGs'**	*loadsharing*
2.3.16	N[n]	**OP'**	PP[of] *engine* **DG'**	*deloading*
2.3.20	V[Pas]	**PMS'**	NP[nom] *the following DGs* **DGs'**	*deloaded*

2.3.21	N[n]	**PMS'**	PP[of] *all ...DGs* **DGs'**	*loadsharing*
2.4.22	V[Fin]	**OP'**	NP[nom] *DGs* **DGs'**	*deloads*
2.4.26 (a)	N[n]		**DG'**	*load*
2.4.26 (b)	V[Fin]	**OP'**	NP[nom] *SG* **SG'**	*deload*

(DE)LOAD

- (de)load; N[n]
- **(de)load'**(ag(x), pat(y))
- ()
- Reading **(de)load'**(ag(x), pat(y))

Ø Ø

E.g. *the (de)load*

(DE)LOADING

- (de)loading; N[ing]
- **(de)load'**(ag(x), pat(y))
- ((<PP[of], [-Anim]>$_1$))
- Reading C_1 == pat(y)

(de)load'(ag(x), pat(y))

Ø (C_1)

E.g. *the (de)loading (of the engine)*

(DE)LOAD

- (de)load; V[act]
- **(de)load'**(ag(x), pat(y))

• (<NP[nom], [+Hum or +Soft]>$_1$, <NP[obj], [-Anim]>$_2$)

• <u>Reading</u> C_1 == pat(y)

(de)load'(ag(x), pat(y))

Ø C_1

E.g. *SG deloads*

<u>LOADSHARING</u>

• loadsharing; N[n]

• **load'**(ag(x), pat(y))

• ((<PP[of], [-Anim, +Plural]>$_1$))

• <u>Reading</u> C_1 == pat(y)

load'(ag(x), pat(y))

Ø (C_1)

E.g. *the loadsharing (of DGs)*

• **(de)load'**(ag(x), pat(y))

x' ∈ {PMS', OP'}
y' ∈ {SG', DG1', DG2', DG3'}

load'(ag(x1), pat(y1)) –> **deload'**(ag(x2), pat(y2))
where (y1' = DG1...n' & y2' = SG') or (y1' = SG' & y2' = DG1...n')

loadsharing –> y' = DG1...n' (n ≥ 2)

<u>Denotations</u>	<u>Linguistic expressions</u>
DGi'	DGs online DGs engine the following DGs all ... DGs

COMMAND'

Occ	GramCat	Agent	Patient	Goal

2.1.15	V[Pas]	PP[by] *the operator* **OP'**	NP[nom] *start and stop of DGs* **DGs'**	NP[nom] *start and stop of DGs* **start and stop'**
2.1.20	N[n]	PP[from] *the ISC consoles* **OP'**	**SG'**	**stop'**
2.1.29	V[Pas]	PP[by] *the operator* **OP'**	NP[nom] *start and stop of SG* **SG'**	NP[nom] *start and stop of SG* **start and stop'**
2.2.6	V[Psp]/ADJ	PP[by] *the PMS system* **PMS'**	NP[nom] *connection of MB to BB* **(DG)MB'**	NP[nom] *connection of MB to BB* **connexion to BB'**
2.4.25	V[Pas]	**OP'**	NP[nom] *the SG* **SG'**	VP[Inf] *to stop* **stop'**

COMMAND

- command; N[n]
- **command'**(ag(x), pat(y), goal(z))
- ((<PP[from], [+Hum or +Soft]>$_1$))
- Reading $\quad C_1 == ag(x)$

$$\begin{array}{l} \textbf{command'}(\text{ag}(x),\ \text{pat}(y),\ \text{goal}(z)) \\ \qquad\quad (C_1) \qquad\ \varnothing \qquad\ \ \varnothing \end{array}$$

E.g. *a command (from the operator)*

COMMAND

- command; N[n]
- **command'**(ag(x), pat(y), goal(z))
- ((<PP[from], [+Hum or +Soft]>$_1$))
- Reading C_1 == ag(x)

command'(ag(x), pat(y), goal(z))

ag(x)	pat(y)	goal(z)
(C_1)	Ø	Ø

E.g. *a command (from the operator)*

COMMANDED

- commanded; V[Pas]
- **command'**(ag(x), pat(y), goal(z))
- (a) ((<PP[by], [+Hum or +Soft]>$_1$), <NP[nom], [-Anim]>$_2$)
- (a) Reading C_1 == ag(x)
  C_2 == pat(y)
  C_2 == goal(z)

command'(ag(x), pat(y), goal(z))

ag(x)	pat(y)	goal(z)
(C_1)	C_2	C_2

E.g. *start and stop of SG is commanded (by the operator)*

- (b) ((<PP[by], [+Hum or +Soft]>$_1$), <NP[nom], [-Anim]>$_2$, (<VP[Inf], [+Action]>$_3$))
- (b) Reading C_1 == ag(x)
  C_2 == pat(y)
  C_3 == goal(z)

command'(ag(x), pat(y), goal(z))

ag(x)	pat(y)	goal(z)
(C_1)	C_2	(C_3)

E.g. *the SG is commanded to stop (by the operator)*

- **command'**(ag(x), pat(y), goal(z))

x' ∈ {PMS', OP'}
y' ∈ {SG', DG1', DG2', DG3'}
z' ∈ {start', stop', connect'}

•	Denotations	Linguistic expressions
	start'/stop'	start and stop to stop
	connect'	connection to BB

BLOCK'

Occ	GramCat	Agent	Patient
2.2.1 (a) and (b)	V[Pas]		NP[nom] *one DG* **DG'**
2.2.19	V[Psp]/ADJ		NP[nom] *theDG* **DG'**
2.2.28	V[Pas]	**alarm'**	NP[nom] *the faulty DG* **DG'**
2.5.2 2.5.4 2.5.5 2.5.6	N[ing]		NP[nom] *start blocking* **start blocking' (of power consumer')**

BLOCKING

• blocking; N[ing]

• **block'**(ag(x), pat(y))

• ((<NP[nom],[-Anim]>1))

• Reading $C_1 == pat(y)$

block'(ag(x), pat(y))

Ø (C$_1$)

E.g. *the (start) blocking*

BLOCKED

• blocked; ADJ

• **block'**(ag(x), pat(y))

• $(\langle NP[nom],[-Anim]\rangle_1)$

• Reading $C_1 == pat(y)$

block'(ag(x), pat(y))

Ø C_1

E.g. *the blocked engine*

BLOCKED

• blocked; V[Pas]

• **block'**(ag(x), pat(y))

• $(\langle NP[nom], [-Anim]\rangle_1,(\langle PP[by], [-Anim]\rangle_2))$

• Reading $C_1 == pat(y)$
$C_2 == ag(x)$

block'(ag(x), pat(y))

(C_2) C_1

E.g. *the DG is blocked (by the alarm)*

• **block'**(ag(x), pat(y))
x' ∈ {alarm'}
y' ∈ {DG1', DG2', DG3', power consumer'}

• only agentless passives

OPEN' vs. CLOSE'

Occ	GramCat	Agent	Patient	
2.1.26	V[Pas]	**PMS'**	NP[nom] *SG MB to BB* **SG MB'** **(to BB')**	*close*
2.3.4	N[ing]	**MSB'**	PP[of] *the MB* **DG MB'**	*close*
2.4.5	N[ing]	**MSB'**	PP[of] *the MB* **SG MB'** **(to BB')**	*close*
2.4.8 2.4.12	V[Pas]	**MSB'**	PP[of] *the SG MB* **SG MB'** **(to BB')**	*close*
2.4.17	N[Base]	NP[nom] *the PMS* **PMS'**	NP[obj] *the thruster(s)* *MB(s)* **ST'/BT' MB'**	*open*
2.4.49	V[Pas]		NP[nom] *SG MB to BT* *respectively ST* **SG MB to** **BT'/to ST'**	*open*

OPENING/CLOSING

- opening/closing; N[ing]
- **open'/close'**(ag(x), pat(y))
- (<PP[of],[-Anim]>$_1$)

• <u>Reading</u> $C_1 == pat(y)$

open'/close'(ag(x), pat(y))

Ø C_1

E.g. *the opening/closing of the SG MB*

<u>OPEN/CLOSE</u>

• open/close; ADJ

• **open'/close'**(ag(x), pat(y))

• (<NP[nom],[+Hum or +Soft]>$_1$, <NP[obj],[-Anim]>$_2$)

• <u>Reading</u> $C_1 == ag(x)$
$C_2 == pat(y)$

open'/close'(ag(x), pat(y))

C_1 C_2

E.g. *the PMS closes SG's MB to BB*

• **open'/close'**(ag(x), pat(y))

x' ∈ {PMS', MSB'}
y' ∈ {SG MB', DG1...n MB', BT/ST MB'}

• mainly agentless passives

RUN'

Occ	GramCat	Agent	Patient	
2.2.3	V[ing]	**PMS'**	NP[nom] *one or more DGs* **DGs'**	
2.2.4	V[ing]	**PMS'**	NP[nom] *no DG* **DG'**	

2.4.40	V[ing]		NP[nom] *thruster* **thruster'**	
2.5.2	N[ing]	**PMS'**	**power consumer'**	*running input*
2.5.4 2.5.8	N[ing]			*running signal*
2.5.7	N[ing]		NP[nom] *consumer running signal* **consumer'**	
2.5.10	V[ing]		NP[nom] *the consumer* **consumer'**	

<u>RUN</u>

• run; V[act]

• **run'**(ag(x), pat(y))

• (<NP[nom],[-Anim]>$_1$)

• <u>Reading</u> C_1 == pat(y)

run'(ag(x), pat(y))

Ø C_1

E.g. *the thruster is running*

<u>RUNNING</u>

• running; ADJ

• **run'**(ag(x), pat(y))

• (<NP[nom],[-Anim]>$_1$)

• <u>Reading</u> C_1 == pat(y)

run'(ag(x), pat(y))

Ø C1

E.g. *the running consumer*

- **running'**(ag(x), pat(y))

x' ∈ {PMS'}
y' ∈ {thrusters, DG1...n', power consumers'}

- progressive form only

MODE'

Occ	GramCat	Agent	Patient	Instrument
2.1.1	N	**PMS'**	PP[for] *DGs... the SG* **DGs' SG'**	
2.1.3	N		**GS'**	NP[nom] *MANUAL mode* **MANUAL'/ AUTOMATIC'/ SEMIAUTOMA TIC'**
2.1.7	N		**DGs'**	
2.1.8 (b)	N		PP[for] *all DGs* **DGs'**	
2.1.9	N		**DGs'**	NP[nom] *AUTO mode* **AUTO'**
2.1.16	N		NP[nom] *DG SEMIAUTOMATIC mode* **DG'**	NP[nom] *DG SEMIAUTOMATIC mode* **SEMIAUTOMA TIC'**

2.1.23	N		NP[nom] *mode SG SEMIAUTOMATIC and SG AUTOMATIC* **SG'**	NP[nom] *mode SG SEMIAUTOMATIC and SG AUTOMATIC* **SEMIAUTOMATIC'/AUTOMATIC'**
...				

MODE

• mode; N

• **mode'**(pat(x), instr(y))

• (a) ((<PP[for],[-Anim]>$_1$))

• (a) <u>Reading</u> C_1 == pat(x)

mode'(pat(x), instr(y))

(C_1) Ø

E.g. *the mode (for DGs)*

• (b) ((<NP[nom],[+Position]>$_1$),(<NP[nom],[-Anim]>$_2$))

• (b) <u>Reading</u> C_1 == instr(y)
C_2 == pat(x)

mode'(pat(x), instr(y))

(C_1) (C_2)

E.g. *the (DG) (AUTOMATIC) mode*

• **mode'**(pat(x), instr(y))

x' ∈ {SG', DG1', DG2', DG3'}
y' ∈ {MANUAL/SEMIAUTOMATIC/AUTOMATIC', MANUAL/AUTO'}

PRIORITY

• priority; N

• **priority'**(pat(x), instr(y))

• ((<ADJ,[+Degree]>$_1$))

• <u>Reading</u> C_1 == instr(y)

priority'(pat(x), instr(y))

Ø (C_1)

E.g. *(high(er)) priority*

<u>PRIORITED</u>

• priorited; ADJ

• **priority'**(pat(x), instr(y))

• $((\langle ADJ,[+Degree]\rangle_1), \langle NP[nom],[-Anim]\rangle_2)$

• <u>Reading</u> C_1 == instr(y)
C_2 == pat(x)

priority'(pat(x), instr(y))

C_2 (C_1)

E.g. *the (highest) priorited DG*

• **priority'**(pat(x), instr(y))

x' ∊ {DG1', DG2', DG3'}
y' ∊ {master'/1sandby'/2standby'}

master/standby sequence' = priority sequence'

online' & (y' = ~master' -> master') -> stops'(x')
stopped' & (y' = master' -> ~master') -> starts'(x')
where x' = DGi'

• 'internal' definition : "The online, PMS controlled DG with highest priority is frequency controlled. This is called the master DG."